In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

University of Alberta

WEB KNOWLEDGE AND INFORMATION VISUALIZATION (WEBKIV)

by

Yonghe NIU

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisisitons et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-87853-8
Our file Notre référence
ISBN: 0-612-87853-8

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

University of Alberta

Library Release Form

Name of Author: Yonghe NIU

Title of Thesis: Web Knowledge and Information Visualization (WebKIV)

Degree: Master of Science

Year this Degree Granted: 2003

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Date: May 10, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Web Knowledge and Information Visualization (WebKIV)** submitted by Yonghe NIU in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. Randy Goebel
Supervisor

Dr. Osmar Zaiane

Dr. C. Donald Heth
External Examiner

Date: May 10, 2007

Abstract

Abstract: we are currently living in a data intense era. As the volume and complexity of the data increase, it becomes almost impossible for people to understand and extract valuable information without using computer aided tools. Information Visualization is one of the tools for presenting data in order to help people to detect patterns by using human recognition capabilities.

As the Internet continues to expand, web sites become more and more complex. Therefore, the problems of how to help users navigate the Internet without getting lost, how to deliver information effectively, and how to extract user interests from web sites are attracting a great deal of attention. Web data visualization is one of tools that can be to extract valuable information from WWW data to improve various web usage applications.

In this research, a radial tree visualization system is constructed to represent web site structure and display both the web access logs and the results of web mining on those logs. Through the Web Knowledge and Information Visualization (WebKIV), a user can filter, zoom, rotate and display web log information as well as the knowledge created by various web mining methods. The WebKIV also provides animation functionality to simulate user behaviors in cyberspace, and the WebKIV can also help web mining researchers evaluate the difference and the effectiveness of patterns extracted by different web mining techniques.

There is no love like a parent's love.

– anonymous

Acknowledgements

I would like to thank all the people who helped me in completing this thesis.

First, I would like to give my sincere thankfulness to my supervisor, Professor Randy Goebel, for his encouragement, guidance and assistance through my graduate studies.

Next, I would like to thank Tong Zheng and Tingshao Zhu. Their comments played a very important role in design and implementing WebKIV system.

Last but certainly not least, I am forever indebted to the love and caring of my family.

Contents

1 Introduction **1**
- 1.1 What is Information Visualization . . . 1
- 1.2 A Large Set of Abstract Information: Web Data . . . 3
- 1.3 Why Web Data Needs Information Visualization . . . 4
- 1.4 What to Visualize? . . . 5

2 Information Visualization **9**
- 2.1 Visualization Design Issues . . . 10
- 2.2 Web Site Visualization . . . 16
- 2.3 Visualizing Topology . . . 17
  - 2.3.1 Technical Background . . . 20
  - 2.3.2 Focus+Context Visualization . . . 23
  - 2.3.3 Zooming and Panning Visualization . . . 34
- 2.4 Web Mining Information Visualization . . . 41
- 2.5 Web Navigation Visualization . . . 46

3 WebKIV System Architecture **50**
- 3.1 Visualization Principles Discussion . . . 50
- 3.2 WebKIV System Architecture . . . 51
  - 3.2.1 Data Collection Phase . . . 51
  - 3.2.2 Data Parsing Phase . . . 55
  - 3.2.3 Data Transformation . . . 57
  - 3.2.4 Data Filtering & Manipulation . . . 59
- 3.3 Web Site Mapping Algorithm . . . 61
- 3.4 Data Operator Details . . . 65
- 3.5 WebKIV Functionality . . . 72
  - 3.5.1 Web Site Structure . . . 73
  - 3.5.2 Web Site Hierarchy . . . 73
  - 3.5.3 Statistics of Web Page Visits . . . 73
  - 3.5.4 Static User Traversal Pattern Display . . . 74

3.5.5 Subtraction of Two Web Data Sets 74
3.5.6 Web User Behaviour Animation 74

4 Experiments **75**
4.1 Data Preparation 75
4.1.1 Data Source 75
4.1.2 Data Format 76
4.1.3 Navigational Compression Models (NCMs) Data 76
4.2 Experiments 80
4.2.1 http://machines.hyperreal.org 80
4.2.2 http://www.cs.ualberta.ca/ 92

5 Summary and Future Work **100**
5.1 Summary 100
5.1.1 Web Site Mapping 100
5.1.2 Visualization System Design 102
5.1.3 Performance 102
5.2 Future Work 103
5.2.1 Visualization Algebra 103
5.2.2 Fisheye Distortion 104
5.2.3 Content Information vs. URLs 104
5.2.4 Surfer Navigation Assistant 105
5.2.5 Other Potential Applications 105
5.3 Conclusion 105

List of Figures

1.1 DNA Structure from http://www.ornl.gov/hgmis 2

2.1 Total number of towers of United States (date not available) 11
2.2 Distribution of ISPs 12
2.3 GARR-B Network Topology 13
2.4 Internet Gold 14
2.5 CyberMap Landmarks 16
2.6 Narcissus 19
2.7 Internet Cartographer 24
2.8 Picture taken by fisheye lens 25
2.9 Fisheye View (a) normal view (b) fisheye view 1 (c) fisheye view 2 . 25
2.10 Fisheye technique application: Document Linear Lens 27
2.11 Fisheye technique application: Document Fisheye Lens 28
2.12 Site Lens for Bestbuy.com-initial view 29
2.13 Site Lens for Bestbuy.com-browsing 30
2.14 Site Manager 31
2.15 Mappuccino fisheye view 32
2.16 Mapuccino tree view 33
2.17 Astra SiteManager 35
2.18 Crowds at web site 36
2.19 Disktree 37
2.20 Disktree Time Tube 38
2.21 Disktree Spreadsheet 39
2.22 Disktree Spreadsheet 40
2.23 Double Decker Plot 42
2.24 Initial display of msnbc.com data using WebCANVAS. 43
2.25 Stratogram 45
2.26 MosaicG 47
2.27 FootPrint 48
2.28 Harmony Internet Explorer 48

3.1 WebKIV System Architecture 52
3.2 Web site structure 61
3.3 Web site tree structure 63
3.4 Radial tree structure 63
3.5 WebKIV System 66
3.6 Identically generated subtrees 67
3.7 Rectangled Zoom-in and Subtree Zoom-in 68
3.8 Pan view of web page visits statistics 69
3.9 Rotation example 71
3.10 Maximum node level setting 72

4.1 Music Machine web page snapshot 79
4.2 Music Machine Web Site Structure 81
4.3 Music Machine visits distribution. 82
4.4 Music Machine Rectangled Zooming 83
4.5 Music Machine hyperlink traversal distribution 85
4.6 Music Machine user browsing patterns related to drum machines . . 87
4.7 Music Machine web usage difference 88
4.8 Music Machine NCM Comparison I 91
4.9 Music Machine NCM Comparison II 91
4.10 Music Machine Web Surfing Animation 93
4.11 Home page of UACS 94
4.12 Web Site Structure of Department of Computing Science, U of Alberta(UACS) October, 2001 95
4.13 UACS Web Site visits distribution 96
4.14 UACS Web Links Usage Distribution 97
4.15 Web Links Usage Distribution 98
4.16 UACS Subgraph of ~chinook 99

5.1 Radial Tree Problem 101

List of Tables

2.1 Association Rules for Double Decker Plot 42

3.1 Examples of Session format 58
3.2 Example of web page visits count 59
3.3 User traversal pattern count 59
3.4 User surfing animation example 60
3.6 Radial tree Parent-Son relationship 62

4.1 Music Machine Web Log File for Jan/99 and Mar/99 77
4.2 UACS Web Log Data 78

Chapter 1

Introduction

1.1 What is Information Visualization

> *Information Visualization is more than a method of computing. It is a process enabling the user to observe, digest and make sense of the information – Nahum Gershon* [24]

At the beginning of the new century, we are living in what has been called the Information Age. Databases are continuously expanding since they first came into existence. The Internet, which connects databases all over the world, is like a huge distributed database system growing at an exponential rate. Information has never been so easy to reach. It is not easy, however, to find useful information because the Internet is so huge and far more complex than we can understand. Visualization is a tool to help us understand these vast abstract data sets.

Visualization techniques have long been used by human beings to help them understand complex problems. From the ancient maps of the world [34] to today's gene map (Figure 1.1), from solar system animation to automobile design, visualization has permeated our lives. In fact, experiments show that people are better at perceptual recognition than cognitive understanding [22].

Visualization is the process of transforming data, information, and knowledge

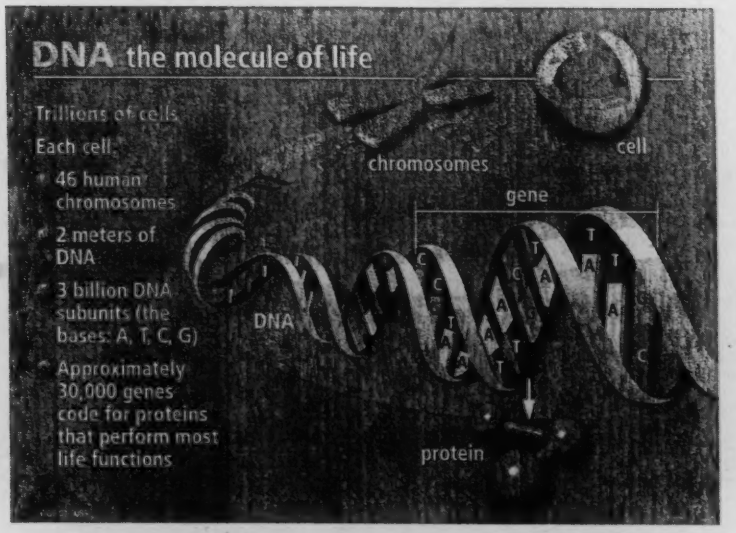


Figure 1.1: DNA Structure from http://www.ornl.gov/hgmis

from abstract information structures into a visual form, in order to make use of human beings' natural perceptual capabilities [25]. Visualization can help people to solve problems like [17][18] :

- Disorientation: graphical representations aim to help users maintain a sense of context within an information space.
- Cognitive overhead: graphical representations can provide an external representation of the user's memory of the navigation session.

While scientific visualization has attracted the interests of scientists who have been dealing with data generated by scientific instruments and simulation equipment, during the last 15 years information visualization has emerged as an independent branch from scientific visualization [49][25]. Some scientific visualization has been developed to interpret data sets such as those of geographical sales distribution which has an inherently spatial character, whereas information visualization deals

with non-spatial data sets, for example, UNIX file directory systems, supermarket transaction databases, etc. To visualize the abstract data sets, mappings have to be developed so that abstract data sets without spatial characteristics can be transformed into a visual space.

Once the data sets have been visualized, a series of interactive visual operations such as zooming, filtering, panning, rotating, and other data manipulating interfaces can be applied to navigate and observe data patterns. Thus, a well-designed interaction operation platform is very important for understanding the information.

1.2 A Large Set of Abstract Information: Web Data

With the information explosion on the World Wide Web, web sites are newly emerged data sets for which the information delivery mechanism is just a mouse click. However, it is inconvenient for web users to find useful information buried in the huge volume of data available on the Internet. How to determine web users's goals to serve them better has drawn lots of researchers' attention recently. Various methods have been applied to try to identify and solve such problems. Search engines are one efficient way to filter out the user's interesting information from the sea of the data sets. However, the accuracy of the end results really depends on the key words the user provides. On the other hand, continually enhancing web site design based on web surfers' behaviour are favored in Internet research community. Some simple statistics-based analyzing software systems which track and analyze the web usage have been introduced in [54]. By analyzing statistics of web log files, these systems try to find "interesting" patterns, e.g., which web page is the most visited, how long a web user spend on a web site, etc. Besides these basic data analyzing tools, web mining, a data mining technique applies on Internet, has gained more interest in the web community [20]. The primary goal of web mining is to discover and analyze useful information from the World Wide Web. The knowledge learned

from web mining should make information delivery more efficiently. For example, different web surfers can access tailored web sites based on their previous surfing behaviour [43, 44]. As well, hyperlinks can be added for "hot" web pages so that those pages are easier to access.

Web Mining can be classified into three categories [20]: Web Content Mining, Web Structure Mining, and Web Usage Mining. Web Content Mining involves extracting information or knowledge from the contents of web pages. Web Structure Mining focuses on discovering and using the web site topology– the relationship among different web pages and that among the contents within a web page. Finally, Web Usage Mining deals with the data describing user behaviour, which is typically recorded in "Web Access logs."

As stated above, web related data includes the web site structures, web site contents, web site logs and web mining resultss.

1.3 Why Web Data Needs Information Visualization

Graphics REVEAL data – Edward R. Tufte

Four factors make visualization systems necessary in order to extract useful information from web data:

1. People are better at understanding colours, lines and shapes than only texts which are inherently abstract [37].

2. The results generated by web mining systems are not suitable for assimilation by human beings. How to help people to interpret the knowledge mined from web usage mining is still a yet to be solved problem. Visualization is a tool to help people to understand web mining data as well as the real world [18].

3. The volume of mined results and the web data is too big to be handled comfortably [30]. For example, data mining researchers always face the tradeoff

between limiting the number of the association rules and omission of useful ones. If we set a higher threshold to get fewer association rules, we might lose some of the important rules. On the other hand, if we set a lower threshold and get more patterns, we face the problem of determining which one is more "interesting" and "valuable" than the others. In addition, a web site itself is complicated. For example, a medium sized web site normally contains 20,000 web pages [49], which would be difficult to navigate without assistance of visualization.

4. By applying visualization techniques to web data, we shift abstract data sets into a physical space. Through a well designed computer-human interface, we can manipulate data sets by using methods such as filtering, zooming and rotating to help us understand and extract "useful" information.

1.4 What to Visualize?

Most recent research papers on web data visualization fall into one of three categories[1]:

1. *Mapping Individual Web Sites* [2]: these systems visualize web site structure, such as the inter-links between web pages and the intra-links within a web page. Both 2D and 3D techniques are used [50, 46, 48, 51, 52, 53, 59].

2. *Web Mining Visualization*: these systems visualize the results of the web mining generated, for example, Association Rules [30] and User clustering [11]. Two methods are used to visualize the mining results. One is to visualize mining results separate from the web site structure [30][11]; Another way is to

[1] There are also many tools to analyze the web log files, getting various statistics numbers such as hits, views and visits. Then the systems use bar charts, pie slices and some other graphics to show the statistics. For more detailed information about web site statistic analysis, please refer to [54]

[2] The web site design visualization tools are out of the scope of this thesis, interested readers can find more information in [34]

apply mining results against the web site structure and display them together as is described in [13].

3. *Navigation Visualization*: navigation visualization is used mainly to help web surfers solve the "lost in the cyberspace" problem [33] so that users can locate what they need and navigate to what they need faster [7, 29, 45, 58, 16, 32, 6, 42].

The goal of the visualization system discussed in this research is to visualize various web data and help users manipulate and observe the data sets so that implicit patterns can be found. Contrary to most of the previous visualization systems that usually visualize web data separately, the present system is unique in that web site structure is displayed as background whereas web content data and web mining data are all superimposed as foreground against web site structure. As a result, the abstraction of the information can be revealed, the interested patterns and trends can "stand out" from the background data and the comparison of the web mining results can be made. Another benefit is that the correlation among the web site structure, web usage data and web content data can be found.

Our research includes the design of a Web Knowledge and Information Visualization (WebKIV) System as well as a prototype implementation. At a conceptual level, WebKIV can be thought of as a set of web data types and a set of operators which transform those data types into images. This formulation provides the basis for achieving the visualization goals for WebKIV, as well as future visualization ideas. For example, both web logs and web structure can be treated as data types, and the visualization of a web structure can be understood in terms of a visualization operator applied with a web crawler to a URL. In WebKIV, a radial tree algorithm [15] is used to generate the web site structure; different web data sets such as user access logs and surfing paths generated by [60] are displayed against the web

site topology; a set of operators are provided as a interactive interface for users to manipulate the web data and emphasize the different aspects of characteristics of same data set. The WebKIV system would help web site designers improve web site design, web researchers better understand web surfers' behaviour and improve information delivery.

WebKIV has the following major functions:

1. Web site mapping: A radial tree algorithm is used to show the web site topology (Figure 3.5).

2. Web user navigation behaviour animation: Most of the web visualization systems show the static results of the web surfers navigation pattern. In WebKIV, an animation technique is used to simulate the web user surfing behaviours in real life.

3. Web mining evaluation: Shows and compares the effectiveness of different web mining applications. Instead of visualizing the web mining results directly, WebKIV generates web logs based on the various web mining results and are put against each other. WebKIV provides a way to evaluate the usefulness of the web mining results. By visualizing the web logs, WebKIV provides a broader way of usage for many of the web mining methods in existence.

4. Web statistics visualization: Shows the web site statistics such as hits, traversal paths, visits and "hot spot" of web site, etc.

5. Showing shifting trends of web usages over time: Multiple views of web usage over times can be displayed in WebKIV so that focus shifting of web users on the web site can be revealed.

6. Revealing the correlation between the web usage and web site design: The influence of the web site changes on web usage patterns can be easily observed.

The rest of this dissertation is organized as follows: Chapter 2 discusses web data visualization related work. Chapter 3 explains WebKIV architecture. Chapter 4 de-

scribes WebKIV experiments. Finally, Chapter 5 presents a discussion and suggests the future work of WebKIV.

Chapter 2

Information Visualization

The term "Information Visualization" was first introduced in Stuart Card's paper "The Information visualizer , an information workspace" [12]. Information visualization was originally applied to database and information retrieval systems [49]. However, as the size and complexity of web sites have increased, web data visualization is needed to help a human manipulate and observe web related data. In [52], the first web data analysis visualization tool was presented. It shows the web log analysis results such as the recency and frequency of access to a web page, by representing them in terms of the thickness, colour of the nodes (web pages) and links (hyperlinks).

Although the system reported in [52] successfully applied visualization techniques to web log data, it is not well developed. For example, the node layout uses a random distribution on the screen. At the time when the Internet first appeared, most web sites were relatively small, i.e., from dozens to hundreds of web pages [47]. But today, web sites often have thousands of web pages (for example, http://www.cs.ualberta.ca/index.html in September 2001 contains over 60,000 web pages not including image files, ps files, etc.). How to display the large amount of data on the screen is the key issue to be considered in visualization. As well, both "what to visualize" and "how to visualize" are other critical questions that need to

be answered. The issue of "what to visualize" involves the visualization of three categories of information spaces: web navigation visualization, web mining information visualization, and web site topology visualization. These three visualization areas will be discussed in details below.

2.1 Visualization Design Issues

While "what to visualize" deals with the data to visualize, the question of "how to visualize" mainly falls in the usage of technology so that the data characteristics can be easily identified through various views. Before delving into the details of the design technology used in WebKIV, which will be discussed in Chapter 3, we first introduce some of the general rules of designing visualization tools. To answer the question "How to visualize", Edward Tufte [55, 56] emphasizes several design strategies for information visualization. His principles can help visualization system designers and are summarized below.

> *Clutter and confusion are failures of design, not attributes of information.[[55] pp51]*

- Micro/Macro Readings. Edward uses the word "Macro" to refer the overview of the information, which delivers to viewers the freedom of choice to compare and sort through details. Micro information provides, on the other hand, the details that accumulate into a coherent larger structure, an overview of the structure. Tufte emphasizes that details should be used to enhance the clarity of the data. "the less subtle the line, the more ambiguous and less interesting is the reading" [[55] pp 51]. All the details of the data (the micro view) combine together to form a Panorama (the macro view). Tufte also clarifies that clutter and information overload result not because of the quantity of details but because of the wrong design. To describe this concept clearer, two exam-

Figure 2.1: Total number of towers of United States (date not available)

ples are provided. Each dot in Figure 2.1 represents a high building which are available to lease for wireless facilities [5]. Over 200,000 dots all together make up a distribution map of high buildings across United States. Figure 2.2, on the other hand, shows ISPs geographical distribution of United States by dying colours to each county according to the number of ISPs onwned by each county [21]. In comparison of Figure 2.1 and Figure 2.2, Figure 2.1 not only shows a good overview, but also shows more details.

Information consists of differences that make a difference. [[55]pp65]

- Layering and Separation. Layering emphasizes the differences, and separation classifies the different attributes of the information by colour, texture, shape, and size, etc. For a high density map, the lack of layering and separation would result in the map being "jumbled up, blurry, incoherent and chaotic" [55]. Information is often multidimensional, so separating the different attributes that are not semantically relevant and layering the information

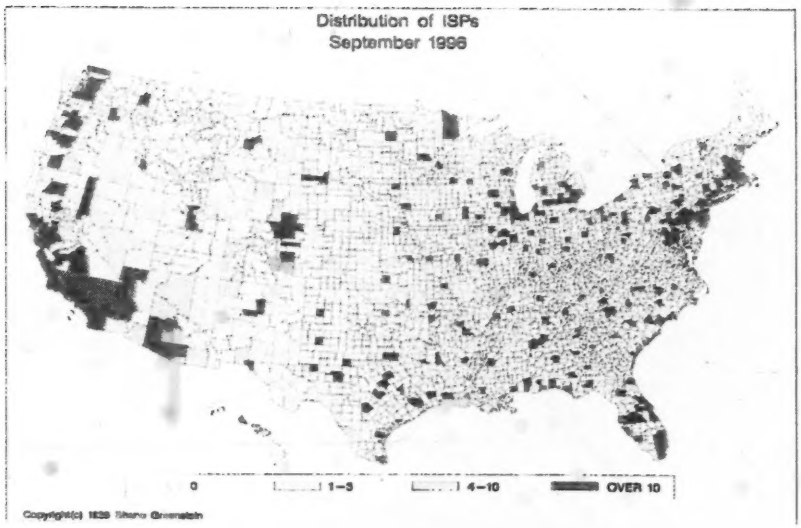


Figure 2.2: Distribution of ISPs
September 1996

for different functions are necessary. For example, one colour or texture can be used to describe the attributes, and another colour or texture can be used for the annotations of these attributes. Figure 2.3 applies this concept well. Different colours of the lines represent the different bandwidth of the networks. Red lines are the backbones of the network across Italy and beige lines are the local connections; White squares and yellow dots are used also to distinguish the hub and satellite cities. The author of Figure 2.3 uses colours as well as shapes to layer the different type of information on one map, making it easy for people to understand. On the contrary, very little layering technique is applied to Figure 2.4. Purple colour lines run all over the graph. Instead of using colour and the thickness of lines to differentiate the differences, viewers have to read the small letters sitting on the lines to tell the difference of the bandwidth [1].

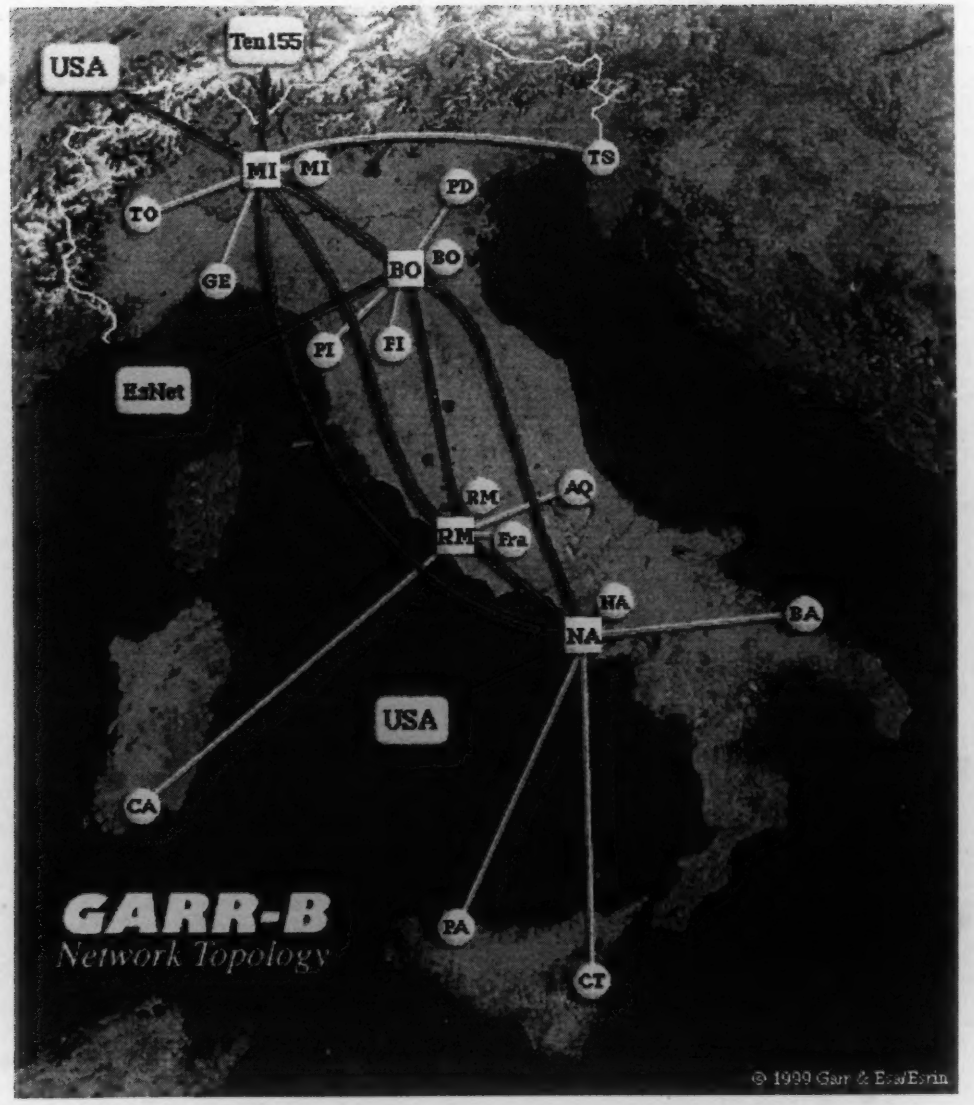


Figure 2.3: GARR-B Network Topology

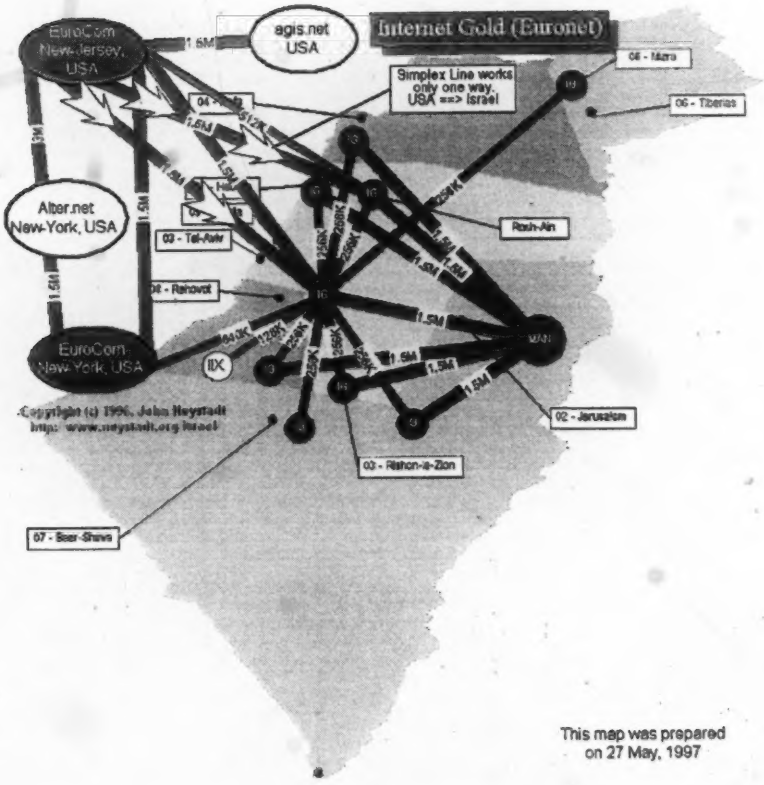


Figure 2.4: Internet Gold

Make all visual distinctions as subtle as possible, but still clear and effective. [[56] pp73]

- Small Multiples. According to Tufte, multiple snap shots of the data are the best way to visually compare the changes, the differences of objects, and the scope of the alternatives. Small multiples should emphasize the changes of data, and the "data slice" should be viewed as a whole without "interrupted visual reasoning." The system designed in [15] is a good example for applying the principle of small multiples. It uses frames of Disktrees to display the difference in the web data along the time variations (Figure 2.21).

Use colours found in nature [56].

- Colour and information. Colour can help readers understand the data. Tufte suggests that, for encoding abstract information, not more than 30 colours be used [[56]pp81] in the same view. As well, nature's colours should be used because they "are familiar and coherent, possessing a widely accepted harmony to the human eye– and their source has a certain definitive authority". Tufte further enlisted some rules when tying colour in the visualization tools.

  1. large area background or base-colours should be neutral so that smaller, important area stand out vividly. The former should be muted, grayish or neutral.
  2. pure, bright, or very strong colours have extraordinary effects if they are used sparingly on or between dull background tones.
  3. large areas of light bright colours with white in between will produce unpleasant results.

A good example to illustrate the importance of tying colour is Figure 2.5 [2]. First, it has a very bright frame which is very distracting. Second, large area of the mustard

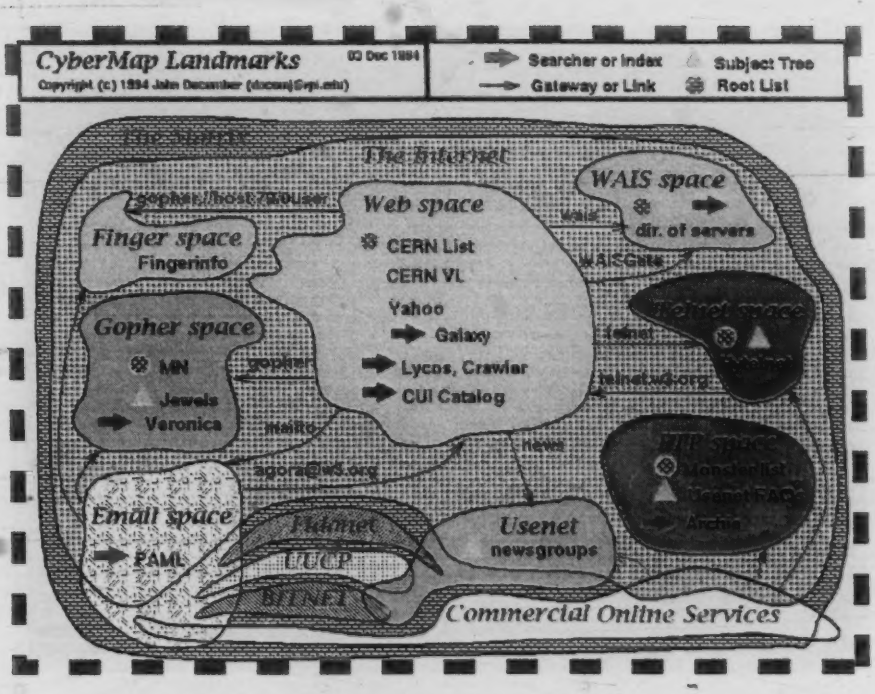


Figure 2.5: CyberMap Landmarks

yellow and turquoise green background mixed with the small pots of the foreground colours, making them hardly stand out to catch the eyes. Finally, the unnecessary dotted and quilt patterns all over the graph further deteriorate the effectiveness of the graph readiness.

The general visualization design principles discussed above are applied to the WebKIV design and used to evaluate the current visualization tools discussed for the rest of this chapter.

2.2 Web Site Visualization

In the context of visualization, a web site is a highly connected cyclic graph where web pages can be treated as nodes in a graph, hyperlinks as lines connecting the

nodes[1]. To visualize a web site, two concerns must be dealt with. One is how to place the nodes in a physical world. The web pages do not have any position preference in nature, but the position for each node must be assigned. Since a web site can easily have over thousands of web pages (for example, the number of html web pages exceeds 60,000 on the web site of Computing Science Department of University of Alberta in September 2001), it is important to choose appropriate algorithms of setting web page position so that the web structure can be revealed, while the clutter and occlusion of the graph can be reduced as much as possible. The other concern is to display as many details as possible while still providing a concise overview of the whole structure.

2.3 Visualizing Topology

The first step to visualize web sites is the structure modelling of web sites. Currently two models are available: hierarchical tree structure and net structure. In the tree model, the home page of the web site is usually treated as the root of the tree, all the web pages which immediately connected by hyperlinks to the home page are treated as the son of the root, and so on. On the other hand, the net structure does not have a parent and son relationship.

This approach argues that web sites were born with a hierarchical attribute. In [38], experiments with the web site design process have been made. The findings showed that the web site designers "generally followed a process of progressive refinement of their designs from less detail to greater detail, and simultaneously from coarse granularity to finer granularity". Simply put, most of the web sites were created in a hierarchical way. The experiments mentioned above indicate that most hand built web sites have a hierarchical structure from the time they were built by

[1]For simplicity, nodes will be used to represent web pages, and lines will be used to represent the hyperlinks connecting two web pages. From now on, they will be used interchangeably in this thesis.

the web designers. Currently the hierarchical tree structure prevails in the visualization community. Most of the web visualization systems adopt this approach [36, 15, 50][47, 45]. However, there are two ways to implement the hierarchical tree structure. One is to use breadth-first search algorithm to generate the web site structure [36, 15, 50]. The structure is generated by a web crawler that starts at a "home page" location and "walks through" the entire web site through the hyperlinks of the web pages. The other implementation is to get the hierarchy of a web site using the file system directory structure where the web pages can be located [47, 45] to determine the web site's structure.

Under the hierarchical approach, the home page is the most important node in the graph, lesser important nodes are the nodes that immediately below the root node and so on. The accuracy of constructing the web site hierarchical tree model using breadth-first search algorithm or file system structure will vary for different web sites. For example, if all the web pages are put in the same directory, the constructed tree of this web site will be very wide and extremely shallow.

While modeling web site as a tree structure is popular in the visualization community, in a strict definition, web sites are not trees. Tree structure is an acyclic graph, where web sites can be either cyclic or acyclic. A simple example, for a web site, is that almost each web page except home page has a "back home" hyperlinks, which forms a circle.

Based on the observation above, some researchers model web sites as a net structure. In a net structure model, there is no root node, every node is equally important. A good example of net modeling of web sites is Narcissus system [26]. Narcissus uses a spring algorithm to place the web pages. In the spring algorithm, nodes repel each other, where links of the nodes act like the springs attracting each other. Node positions is set when the whole system has the minimum energy. Different forces applied to the links generate different views of web pages. The web

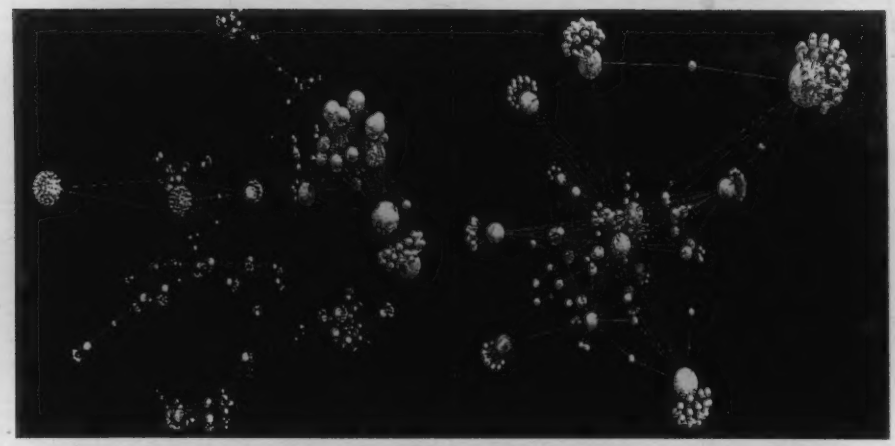


Figure 2.6: Narcissus
Left: Link Force **Right:** Usage Force

page positions are not fixed in the graph, so they can be changed drastically. In Figure 2.6, the left graph shows web page positions in Narcissus according to the hyperlinks they have. The more hyperlinks the web pages have for each other, the closer the nodes will be displayed in the graph. When the forces change to the usage of hyperlinks, the web page positions change as shown in the right graph of Figure 2.6. In this case, when rules of usage of hyperlinks applies to the web pages, the heavier the hyperlinks being used, the stronger the attractive force being applied to the web pages they linked, thus, the closer the web pages.

The spring algorithm used in Narcissus treats each web page equal and no web page position can be predefined. The position of web pages will change when the different forces apply to the web pages. The benefit of spring algorithm is that the number of crossing links is reduced by putting highly connected nodes closer. The problem, however, is that if the web site is connected condensely, line crossing is still inevitable. In addition, the graph changes as the structures change, so comparing the difference for the data sets over time is difficult.

2.3.1 Technical Background

The previous section discussed what topology model to choose to visualize a web site. In this section, we will investigate how to visualize the web site details. It is always a challenge for researchers to visualize thousands of web pages and hundreds of thousands of hyperlinks on a monitor, while still providing both usability and viewability.

The visualization systems should provide an overview of the web site, but an overview cannot show all the web site information. Even if all the information could be displayed, it would be too small to read. Thus, the visualization tools also should have other functions to display the details, like the ability to drill down the information in detail. To fulfill these requirements, four possible visualization techniques are currently available:

1. Abstraction
2. Zooming and Panning
3. Distortion and
4. Multiple Views.

One option of abstraction is to use granularity coarser than that of the web page, or in other words, to use abstraction to reduce the information density. For example, one could group all the web pages which have the same theme (content-based clustering) together and display them as one node. For highly structured data sets such as object-oriented Java programs, abstraction (structure-based clustering) is a good way to display information that is both usable and viewable. Unfortunately, a web site structure is a loosely structured data set. So there is no easy answer for the questions such as "Which nodes can be collapsed together?" or "Can nodes be clustered disjointly and accurately?" Besides, content-based clustering requires

domain-specific data and knowledge which are "so specialized to a problem domain that it is no longer general enough for use in other application areas" [27]. Thus, automatic abstraction (content-based clustering) is either impossible or not accurate. In addition, some of the patterns which we might be interested in are very likely to be lost after the collapsing. From the pattern observation point of view, unless the above two questions can be answered, abstraction visualization is not a good option to map web sites.

Zooming and Panning are commonly used tools in visualization systems, and are similar to the tools used in making maps. For example, the map of a country provides only the most important information on it (the information depends on the purpose of the map). For people who require more details, each city has its own local map. The local map would show more detailed information about the city than the country map would provide. If still more detail information is needed, an area map, like a national park map, is available. It gives the most detailed information about everything from camping sites to scenery. The way that we use different kinds of maps is the same as how we explore a web site. An overview provides only important information, or the main patterns. To show the detailed information for a subset of web pages, a zoom can be used.

The problem with zooming and panning is that it provides detailed information in a local area but loses the overall view of the whole structure and the correlation with related information.

The idea of a Distorted Display has emerged as an alternative to zooming and panning. It can show detailed information in a particular area without eliminating information about the context. Two terms are used for the distortion techniques: "Focus" and "Context". "Focus" refers to the local area of a graph that shows more details than the rest of the area of the graph. The area is semantically blown up when this area is the focus. "Context" means the rest of the area of the graph. The

context area is reduced to display less information than was displayed before the reduction. Therefore, a focus + context visualization technique magnifies a small area of a display and reduces the area of the rest of the display to provide both focus (detailed) information and context (overview) information (see Figure 2.12). The most popular and full-blown techniques are fish eye view [23] and hyperbolic view [36], which will be discussed later.

The most attractive attribute of a distorted display is that it provides detailed information and some of the neighbouring data information. Using this display is an efficient way to solve the "lost in hyper space" problem [33]. Although distorted display does not have the semantic switching disadvantage that is inherent in zooming and panning techniques, it is not effective for data analysis or pattern extraction. It cannot give a good overview simply because it "distorted" some areas of the graph. Thus, observing the patterns that may cross the web site is difficult. Moreover, their inherently complex geometry could make them difficult for viewers to understand. Though theoretically, an infinite number of objects can be put into a space, the distant information from the "focus" becomes so small that viewers can barely see it [40].

Multiple view systems use two or more views to display the different aspects of the same data set [8]. Multiple views can benefit the users in several ways: First, multiple views minimize the information overhead. For example, some of the data sets are high-dimensional, so to display them all at once would simply puzzle viewers. Decomposing the complex data into relatively simple data would increase the user's understanding. Second, since data can have various attributes, there is no need to display all the less relevant attributes together. Third, multiple views can help users to understand complex relationship among different data sets, for example, data sets collected from different times. Finally, some patterns are extractable only when viewed from different angles or when different data sets are

put together.

Though the multiple view technique has the advantages stated above, it has its own disadvantages. First, multiple views consume valuable display space. Second, this technique requires extra computation. Multiple views require complicated synchronization of different views, which always involves a great deal of computation such as coordination transformation and data sets update. Last, multiple views can create cognitive overload. They require the user to pay more attention to context switching from one view to another. Thus, the multiple view technique must be used very carefully. Though the disadvantages listed above exist, they are not inevitable, because they can be alleviated to some degree. For example, in [3], to solve the problem of losing the correlation with the context data with zoom, the multiple view technique can be used to display both the overview and the enlarged view as shown in Figure 2.7.

2.3.2 Focus+Context Visualization

In this section, several visualization systems are discussed, to illustrate the usage of Focus + Context technique.

2.3.2.1 Fisheye Visualization Introduction

In photography, an extremely wide-angle camera lens is called a "Fisheye Lens". This lens makes reality appear distorted in pictures taken close up. A characteristic of photos taken with this lens is that objects in the center of the picture appear larger while objects at the edge appear smaller (Figure 2.8). The fisheye view visualization technique is a kind of simulation of the wide-angle lens of cameras.

The fisheye view was first generalized in [23]. It shows the center area of the image in great detail and the edge areas of the image in less detail (Figure 2.9)

The fisheye view has been implemented and applied by several visualization tools [9, 53, 40] in the early 1990's. The fisheye effect can be achieved by using a

Figure 2.7: Internet Cartographer
The left window is for the zoom; the right bottom window is for the overview.

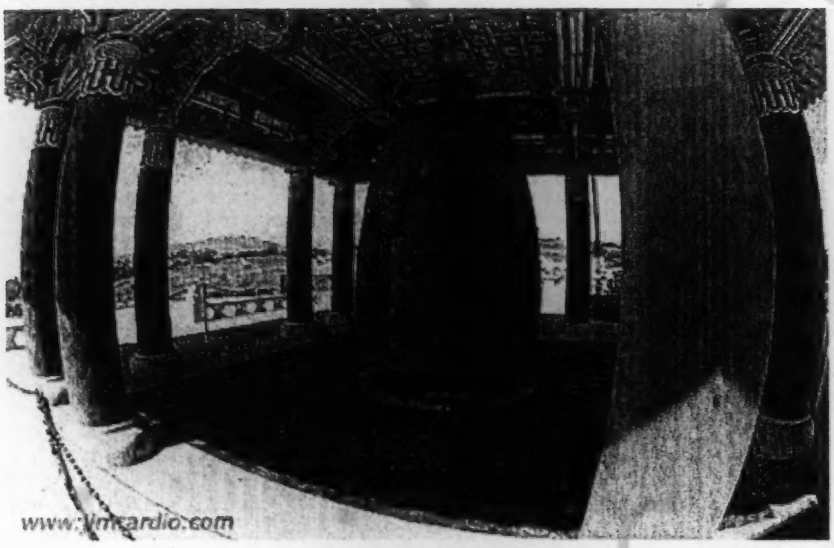


Figure 2.8: Picture taken by fisheye lens

Figure 2.9: Fisheye View (a) normal view (b) fisheye view 1 (c) fisheye view 2

simple geometric transformations, rather than by implementing a fisheye model. It can be used with most of the non-distortion visualization systems that have already been designed and used [35] so that the fisheye effect can be shown. A example is [31], in which the authors apply the fisheye technique to test the electronic document reading preference along with other two interfaces: linear and multiple view magnification (Figure 2.11). The authors conclude that multiple view reading is the best choice for usability whereas the fisheye effect is suitable for time-critical readings.

2.3.2.2 Hyperbolic View

The hyperbolic view was first introduced in [36]. The effect is similar to that of the fisheye view. A hyperbolic view lays out a graph in a hyperbolic geometry and then projects it onto a Euclidean plane or space to generate either 2D or 3D graphs [27]. Several implementations of hyperbolic technique are discussed below.

- Site Lens of Inxight

The site lens of Inxight applies the hyperbolic technique described in [36] in a 2D plane. A web crawler is used to get the web site structure. A breadth-first search algorithm is applied by starting from the homepage to the boundary of the web site (within the root page domain).

In the site lens, the web pages can be moved into focus by clicking and dragging the node rectangle to the center; thus, the node representing the web page will be enlarged, and the other nodes will be pushed to the periphery area where they are shunk. By continuing to drag nodes into focus, the whole web site structure can be explored. Double clicking the node on the map pop up a web browser that displays the web page. Another function of Inxight is to show the search result on the graph by marking the matching nodes in different colours.

Executable Object Modeling with Statecharts
David Harel & Eran Gery, Computer, JULY 1997, 30 no. 7, 31-42

Statecharts, popular for modeling system behavior in the structural analysis paradigm, are part of a fully executable language set for modeling object-oriented systems. The languages form the core of the emerging Unified Modeling Language.

Models for the development of object-oriented systems should be behaviorally expressive and rigorous as well as intuitive and well structured. Thus, any modeling approach must be detailed and precise enough to produce fully executable models and permit the automatic synthesis of efficient code in languages such as C++.

Most OO modeling methodologies specify a model through graphical notations. Entity-relationship-like diagrams typically specify object classes and their interrelationships, and there is some way to describe what objects do and how they interact. Most methodologies also adopt a state-based formalism to specify behavior, using statecharts[1] or some sublanguage thereof.

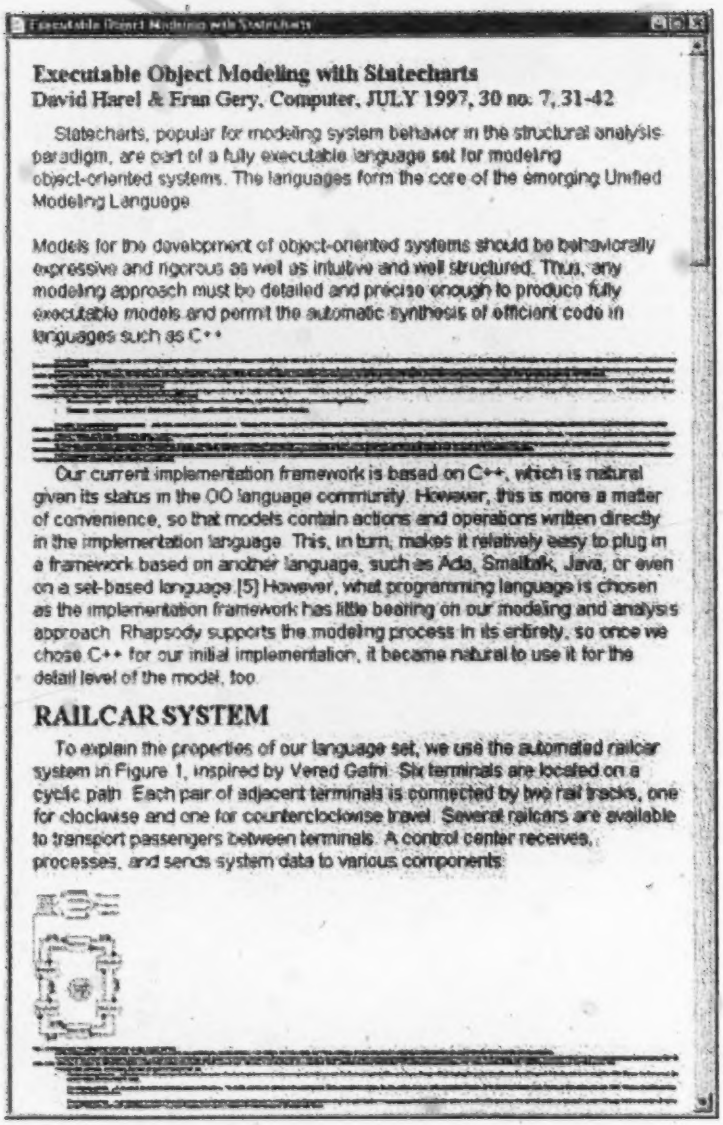
However, many methodologies fail to rigorously define the semantics of the languages. Without a rigorous semantic definition, precise model behavior over time is not well defined and full executability and automatic code synthesis is impossible. Adopting a richly expressive behavioral language like statecharts makes modeling easier, but requires great care in defining the way it integrates with the other parts of the model. Statecharts must capture not only the state of the object as a precondition to service requests, but also the dynamics of the object's internal behavior in responding to those requests and in maintaining relationships with other objects.

These issues are complicated and go beyond recommending a modeling approach or methodology—they are language design concerns, requiring rigorous mathematical underpinnings. Both syntax and semantics must be fully worked out, any possible combination of constructs must be clearly characterized as syntactically legal or illegal, and each legal combination must be given a unique and formal meaning.

To address these needs, we embarked on an effort to develop an integrated set of diagrammatic languages for object modeling, built around statecharts, and to construct a supporting tool that produces a fully executable model and allows automatic code synthesis. The language set includes two *constructive* modeling languages (languages containing the information needed to execute the model or translate it into executable code).

- *Object-model diagrams* specify system structure by identifying object classes and their multiplicities, object relationships and roles, and subclassing relationships.
- *Statecharts* describe system behavior. A statechart attached to a class specifies all behavioral aspects of the objects in that class.

Figure 2.10: Fisheye technique application: Document Linear Lens



Executable Object Modeling with Statecharts
David Harel & Eran Gery, Computer, JULY 1997, 30 no. 7, 31-42

Statecharts, popular for modeling system behavior in the structural analysis paradigm, are part of a fully executable language set for modeling object-oriented systems. The languages form the core of the emerging Unified Modeling Language

Models for the development of object-oriented systems should be behaviorally expressive and rigorous as well as intuitive and well structured. Thus, any modeling approach must be detailed and precise enough to produce fully executable models and permit the automatic synthesis of efficient code in languages such as C++

Our current implementation framework is based on C++, which is natural given its status in the OO language community. However, this is more a matter of convenience, so that models contain actions and operations written directly in the implementation language. This, in turn, makes it relatively easy to plug in a framework based on another language, such as Ada, Smalltalk, Java, or even on a set-based language [5] However, what programming language is chosen as the implementation framework has little bearing on our modeling and analysis approach. Rhapsody supports the modeling process in its entirety, so once we chose C++ for our initial implementation, it became natural to use it for the detail level of the model, too.

RAILCAR SYSTEM

To explain the properties of our language set, we use the automated railcar system in Figure 1, inspired by Vered Gafni. Six terminals are located on a cyclic path. Each pair of adjacent terminals is connected by two rail tracks, one for clockwise and one for counterclockwise travel. Several railcars are available to transport passengers between terminals. A control center receives, processes, and sends system data to various components.

Figure 2.11: Fisheye technique application: Document Fisheye Lens

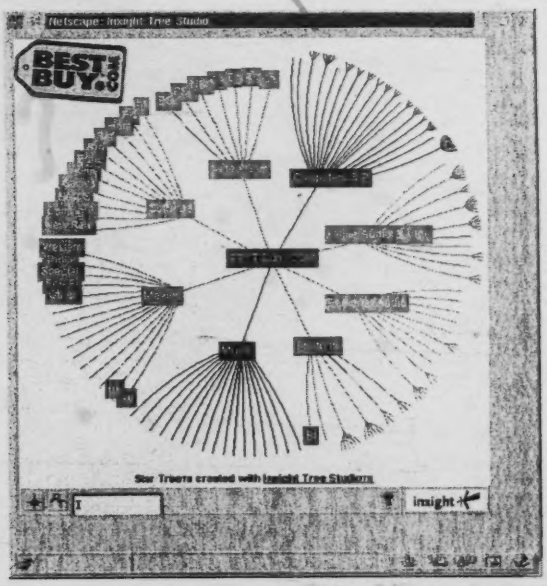


Figure 2.12: Site Lens for Bestbuy.com-initial view

The site lens is ideal for web site navigation; users can compare the context information before they choose a hyperlink to drill down (Figure 2.12). Another advantage is that when searching for something, users can very easily find the search results by viewing the density of the marks (Figure 2.13).

The problem with a hyperbolic tree is that in a big web site, web users still can easily lose their orientation. Since it will not provide an overview for the entire web site topology, comparing the patterns put against it is difficult.

- SiteManager

In [50, 48, 47], the hyperbolic technique was used in 3D. Site manager provides the same function of exploring the web site structure as Site lens of Inxight [36], such as the ability to shift different web pages away from and into focus (Figure 2.14). It is designed for web masters and content designers to provide a fluid view of the web site structure. The advantage of Site Manager is that a 3D graph is more

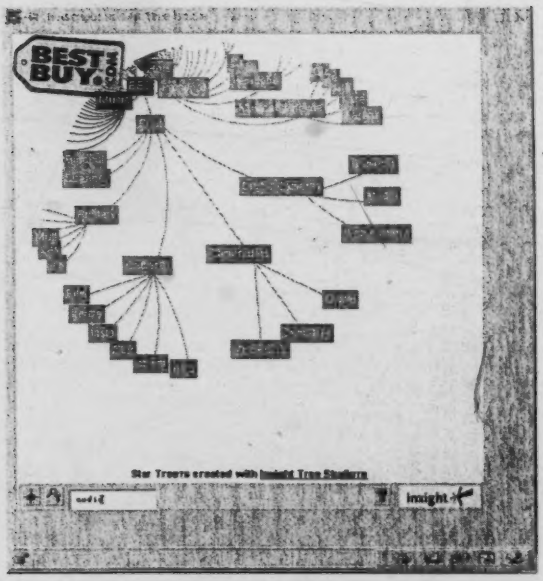


Figure 2.13: Site Lens for Bestbuy.com-browsing

like human beings perceive the real world. Besides, a 3D image also provides more space than a 2D image for a large amount of information to be displayed. However, 3D visualization has its own problems. First, it easily occludes. Second, crossing of the links is very likely to happen. Finally, people perceive a 3D object displayed on a 2D screen as "unnatural."

- Mapuccino

While most of the web site visualization tools visualize the whole web site, Mapuccino (WebCutter) displays only part of the web site structure based on the user's "interests" [41]. Behind Mapuccino, there is a search engine [28]. Mapuccino displays and highlights the web pages which contains the search string and the paths via which the user can reach the "interested web pages." The web pages which are not the "interests" or not on the route to the web pages which contains the searched string is "cut off" from the graph and not displayed. In order to give a better view

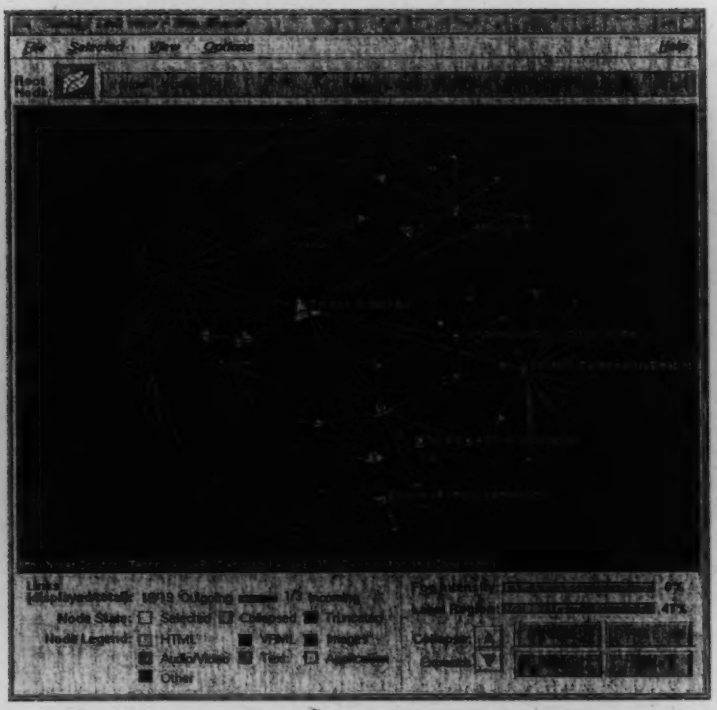


Figure 2.14: Site Manager

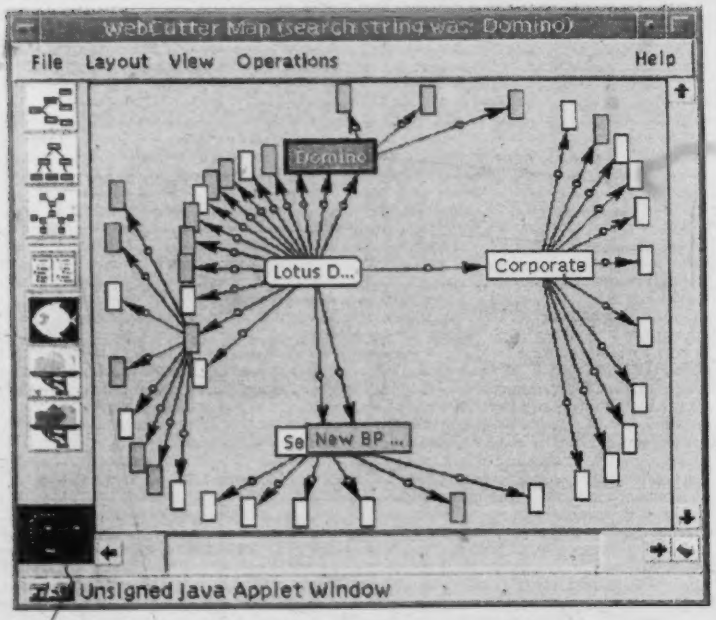


Figure 2.15: Mappuccino fisheye view

of the search results, several visualization techniques are used in WebCutter, for example, a tree view (Figure 2.16) and a fisheye view (Figure 2.15). Since Mapuccino generates the web site in real time, performance is always a concern. As well, it never generates an overview of the whole web site.

Although the distortion display technique has its own advantages, e.g., the ability to show the context information in less details and to focus information in more details, it cannot give the user an overview of the whole data set, and especially for a large web site, dragging and moving are inevitable. In contrast, Zooming and Panning does not have those problems.

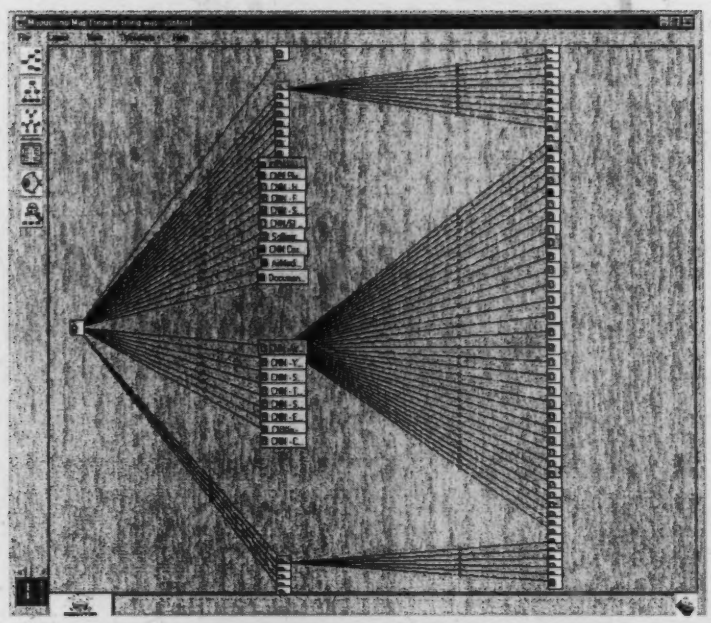


Figure 2.16: Mapuccino tree view

2.3.3 Zooming and Panning Visualization

Zooming and Panning technology does not need to "distort" the map to enlarge part of the graph and shrink the rest. Instead, it provides the overview of the whole web site as well as the detail view. In this section, we will review some of visualization tools using zooming and panning technologies. Note, zooming and panning can be used in most visualization tools.

- Astra SiteManager

Astra SiteManager [4] is a visualization application that uses zooming and panning technology to display a web site. Two views as shown in Figure 2.17 are available in SiteManager : a small overview map for panning and a detail view for zooming to show the detail information of the part selected from the overview window. In SiteManager, a 2-D cone tree layout (Figure 2.17) is used to display web pages and links. The parent web page is placed at the apex of a cone, the children of the web page are laid out along a circle, with the space occupied by each web page being proportional to the number of children it contains.

One useful function of Astra SiteManager is its ability to overlay usage patterns against the site structure. The ability to show how a site is being used as well as its structure is a potentially very interesting aspect of mapping web sites. This ability provides very useful information for the web designers. Consider Figure 2.17, where the red colour represents the "hot" web pages users are interested in, and the blue colour represents the pages that are scarcely visited by the web surfers. The ability to combine information structure and user activity in a single map creates a deeper understanding of the geography of a web space.

- Crowds Dynamics

A web site is treated as an inhabited, social, active space in Minar's Crowds Dynamics [42]. The author implements an animation visualization system to illus-

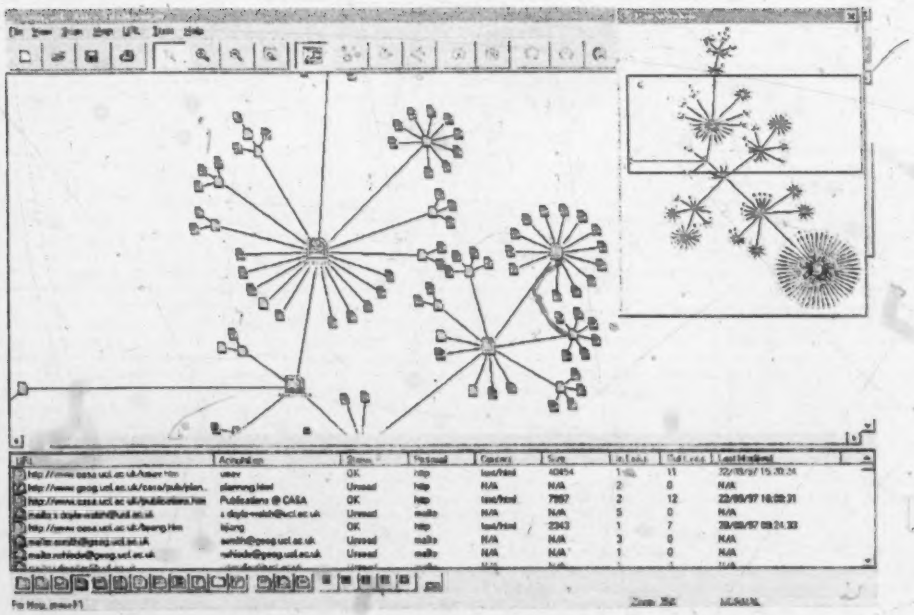


Figure 2.17: Astra SiteManager

trate how the web users walk, loiter in the web, which part of the web site is more popular, which paths the users like to use, etc.

To visualize web users' activity, a web site space is constructed. Each rectangle icon represents a group of web pages that are semantically relevant (handmade). The position of each node is assigned according to the web site's organization. Each web user is represented by a dot coloured according to his or her top-level domain name (Figure 2.18). For example, red could be used for a user from the .edu domain. To illustrate the popularity of the web pages, the recently visited rectangle turns bright, whereas the rarely visited rectangles fade into the background. For user activities, dots move from one rectangle to another as users click the hyperlinks. The dot fades out until it disappears if the users do not have any information recorded in the web log file during a fixed period of time. Such users are assumed to have left the web site.

Figure 2.18: Crowds at web site

Crowd dynamics animation is simple but information-rich. Most of the visualization systems show the user activities in a static way, but [42] is like a documentary movie and revives the real-time user behaviour along the time lines.

The system is informative but too simple, many improvements need to be made. First, the web site structure is handmade, web site specific. To automatically generate a web site map is a major challenge. For example, one needs to know how to group web pages and how to place them. Second, the speed of animation in a big web site map is critical to the success of the animation. Usually, people treat the movement as continuous if the frame rate is less than 200ms [48]. A very careful design for a large web site's animation is required. Last, rather than showing "what people are doing", the system would be better if users' activities were recorded in an aggregate way. For example, to show "what people have done over time" reveals more information.

- Disktree

Figure 2.19: Disktree
The concentric circles represent the hierarchy of the web pages in Disktree

The Disktree visualization system was developed by Ed H. Chi et al. at Xerox Palo Alto Research Center. Based on the Disktree system, several experiments were done to show the web usage pattern over time, web site structure evolution and web content clustering.

In [15], a disktree (Figure 2.19) is developed to discover trends in the evolution of web site usage and web site modification. A disktree is a concentric circular hierarchical layout in which the root node is in the middle of concentric circles. Each inner circle's nodes are the parents of the immediate outer circle's nodes. A breadth-first search algorithm is used to build the web site structure. "Breadth-first traversal transforms the web graph into a tree by placing a node as closely to the root node as possible" [15]. The links represent the tree structures, as is shown in

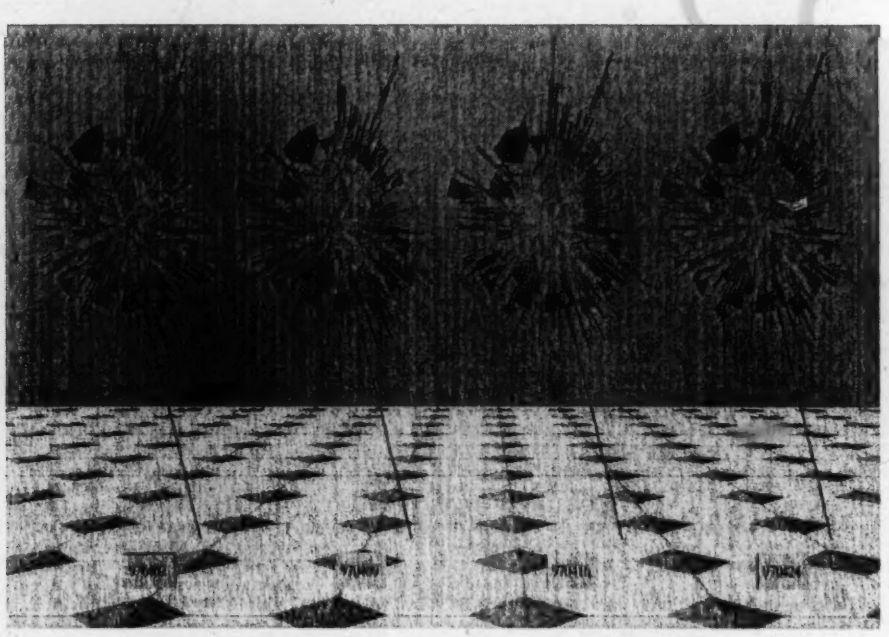


Figure 2.20: Disktree Time Tube

Figure 2.19. Each line in a disktree bears the node information, e.g., the thickness and brightness of a line correspond to the node's visit frequency. The thicker and brighter the lines, the more the page has been visited. Colours are also used to represent the node information (e.g., red is for a new added page; yellow is for a deleted page). To present evolving trends, several disktrees with different web log files over time were made and aligned together, conforming a time tube, as is shown in Figure 2.20. Via the Time Tube, viewers can observe the popularity shift over time, the correlation of the new added and/or deleted web pages and the usage patterns, etc.

A further extension has been made based on the disktree visualization system. In [14], a multiple view visualization system has been built, to evaluate web usage patterns and web usage evolvement over time (Figure 2.21). If viewers are interested in specific patterns, a drill down interface is provided to view the detailed

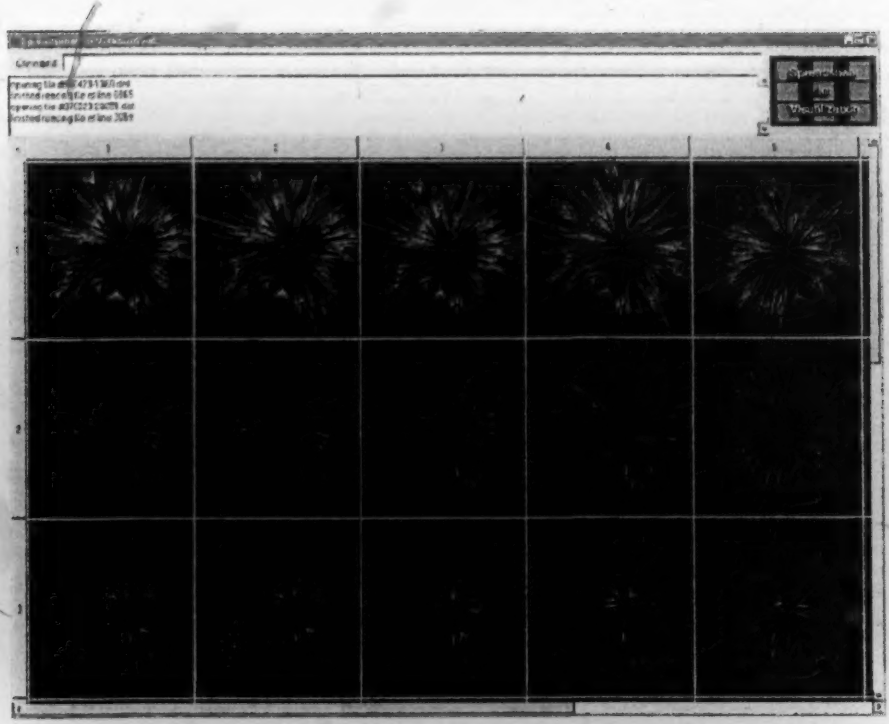


Figure 2.21: Disktree Spreadsheet
Different thresholds show the pattern discovered over time. Each column represents one week's usage data. The X-axis is for time, the Y-axis is for the web access frequency threshold. Row 1 is from 0 to 100, row 2 is from 100-500 and row 3 is from 500 to 2000.

information, as is shown in Figure 2.22 row 2.

While most of the visualization system discussed above deal with the structure, content and usage in isolation, Disktree is the first visualization system to visualize web site usage, web site content and web site structure all together. Also worth mentioning is that efficient usage of the two operators to highlight the useful patterns: colour scale thresholds setting and data subtraction. By setting different scale thresholds of colours in the visualization system, information can be layered so that interested information can be emphasized on demand whereas rest of the data can

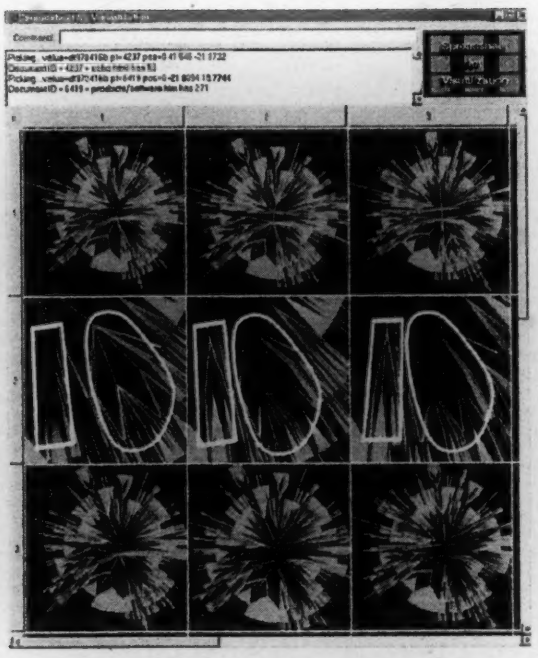


Figure 2.22: Disktree Spreadsheet

be faded out or even hidden as is shown in Figure 2.21. Subtraction in the disktree visualization system is another powerful operator to emphasize the difference over different data sets. For example, in Figure 2.22, Column one shows the result of subtracting week one from week two, while Column two represents the result of week three subtracted from week two, and so on. Thus, over time, the part visited more and the part that has lost popularity are clearly shown (in Row 3, blue represents the negative values, red corresponds to the positive values).

Disktree has several advantages for data driven, pattern extraction tasks. First, the structure of the tree is visualized compactly, with the pattern easily recognizable. Second, the algorithm is simple to understand. Third, no occlusion problem exists since Disktree is on a 2D plane. Lastly, with the most visited web pages shown in the inner circles, the author claim that it is "aesthetically pleasing to the eye." [15]

2.4 Web Mining Information Visualization

Web mining systems can extract patterns and rules from large data sets, but the number of patterns and rules can be huge [39] and they are not easy for human beings to interpret and understand [30]. Visualization can help people to filter and find "useful, valuable" patterns as well as to understand the patterns mined.

For example, in [30], a Mosaic plots technique is used to visualize association rules. In double decker, the Y-axis (vertical) of the graph represents the confidence whereas X-axis (horizontal) represents the support of the association rule. Consider the example of Table 2.1 as shown in Figure 2.23. Each thin bar at the bottom of the graph represents one product: *corned beef, avocado and artichoke* respectively. There are 8 different combinations of the 3 products in total ({}, {*artichoke*},{*avocado*}, {*avocado, artichoke*}, {*corned beef*}, {*corned beef, artichoke*}, {*corned beef, avocado*}, {*corned beef, avocado, artichoke*}). From the graph we can tell: 1) the customers who buy *artichoke and avocado* are very likely

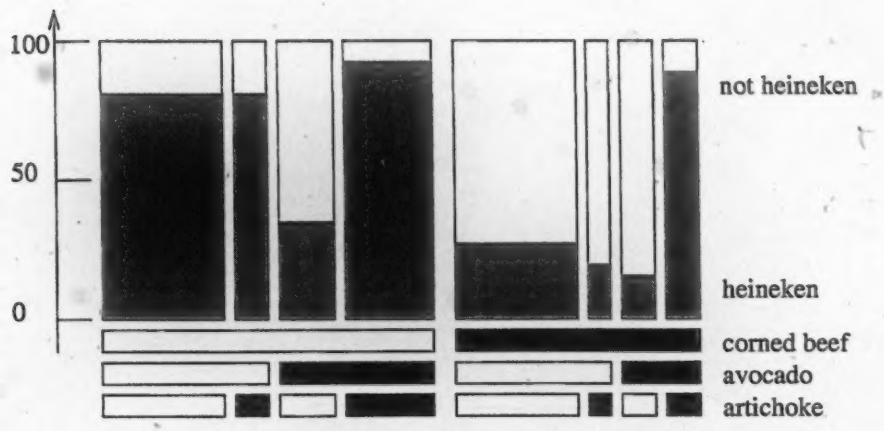


Figure 2.23: Double Decker Plot
The 3 bars at the bottom represents the combination of the three products

Association Rule	confidence	support
avocados & artichokes - > Heineken	95	15.7
corned beef & avocados & artichokes - > Heineken	89.74	3.50

Table 2.1: Association Rules for Double Decker Plot

to buy *Heineken* (The fourth bar from left represents a high confidence (height) and relative high support (width)). 2) The customers who buy *artichoke, avocado and corned beef* are not likely to buy *Heineken* (The eighth bar is very slim representing very low support). These findings implied that *"avocado & artichoke -> Heineken"* is a good rule where as *"avocado & artichoke & corned beef -> Heineken"* does not.

While the mosaic plot described above is a good start to help users understand the individual association rules, it can not filter out the "good" rules from large amount of association rules.

Besides association rules, web user clustering is another important pattern to visualize. The research of finding common user surfing behaviour and grouping

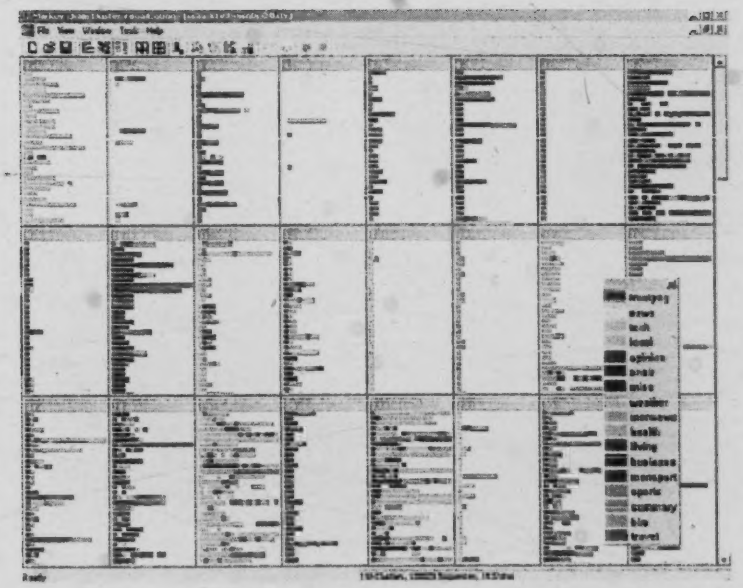


Figure 2.24: Initial display of msnbc.com data using WebCANVAS. Each window corresponds to a cluster. Each row in a window corresponds to the path of a single user through the site. Each path is colour coded by category. The category legend is at the lower right of the screen.

them together makes adaptive web sites possible. An adaptable web site can provide information based on user interests so that information can be delivered more efficiently [43, 44]. In [11], a visualization tool called webCANVAS (Web Clustering ANalysis and VisuAlization of Sequences) is developed to display the user clustering information. In Figure 2.24, all the web pages of the msnbc.com web site are classified into one of 18 categories and each category is assigned a different colour. Each small window in the graph represents a user cluster. Each bar in the small windows represents one user's entire traversal session. The square on the bar represents one of the web pages that the user visited. The colour tells which category the web pages belongs to. In cluster 12, most viewers of this group who go "on air" news will also go to "local news", and in the same sequence. With the aid of WebCANVAS, common user surfing patterns can be easily identified and extracted. The problem, however, is that the screen space of WebCANVAS is not used efficiently. The example given above contains 100,023 user traversal sequences and 100 clusters. For each cluster window, an average number of users will exceed 1000, whereas each cluster window can display only dozens of user surfing patterns. Continuous scroll back and forth is inevitable.

The stratograms of Berendt [10] are also designed to provide a visual basis for navigation analysis. The visualizations can be scaled to multiple levels of detail, and have been deployed to interpret the browsing behaviour of e-commerce customers. They have been successfully used to construct high level descriptions of customer behaviour (e.g.,"communication phase," "information phase"). Stratograms can not provide any insight into web site structure, as they are designed for use in focused applications. They are also designed to interpret sequential patterns of both individual and aggregate users, so their use for web site redesign or navigation compression would be limited.

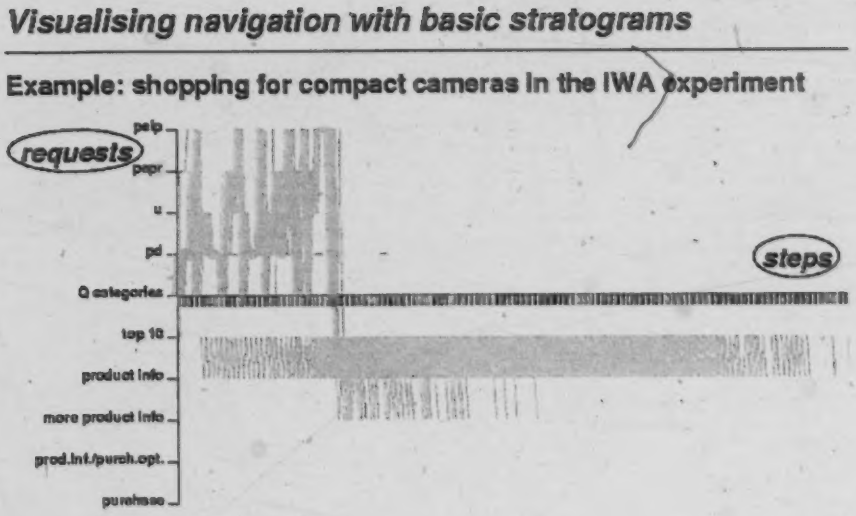


Figure 2.25: Stratogram

2.5 Web Navigation Visualization

To deal with the problem that many papers have discussed as being "lost in cyber space" [33], various navigation assistance systems have been developed during the last few years [52, 16, 58, 6, 45, 7, 29]. These tools all have a basic character: they visualize only the web pages and hyperlinks that the user visited.

In MosaicG [7], a hierarchical tree is designed to show the history of user visits (Figure 2.26). For each page the user visited, a thumbnail picture to reveal the content of the web pages is added to the tree view. Subtree zoom in/out operations are applied so that the user can choose to view either the overview of the visited web pages or the detailed information on each page. The system is merely an individual user traversal recorder. No aggregate information is provided to benefit the subsequent users.

Another visualization tool FootPrint described in [58, 57] focuses more on the value of the user experience in a web site. Foot print applies a hyperbolic tree technique to visualize the user traversal patterns (Figure 2.27). The tree shows the well-used hyperlinks and the web pages that were frequently visited by the previous users. The concept of the "FootPrint" is based on the idea that the interaction history with the web site and the work done by previous users can help future users to navigate in a complex web site. In this way, the individual surfing activity is correlated to that of other web users. The visiting patterns (foot prints) the previous users left are aggregated and saved for references for the new web surfers.

Whereas the above two systems are history-focused, Harmony Internet Explorer [6] is "future" oriented. To solve the lost context problem, it provides users with the context information of the web page that the user has just arrived at (Figure 2.28). It works like a local web crawler, automatically searching the fixed depth of web pages rooted from the page that the user just arrived at and then displaying them. Based on the local map, users can more easily decide their next move than they

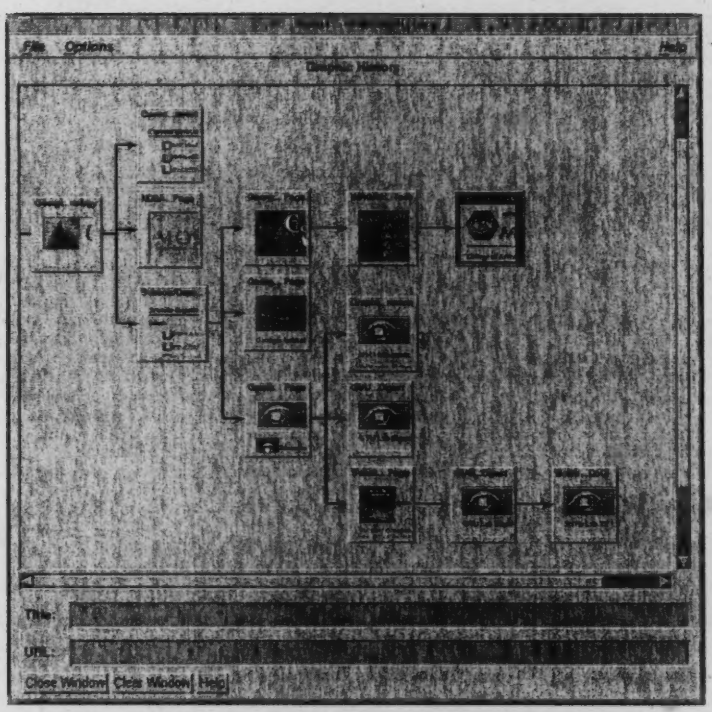


Figure 2.26: MosaicG

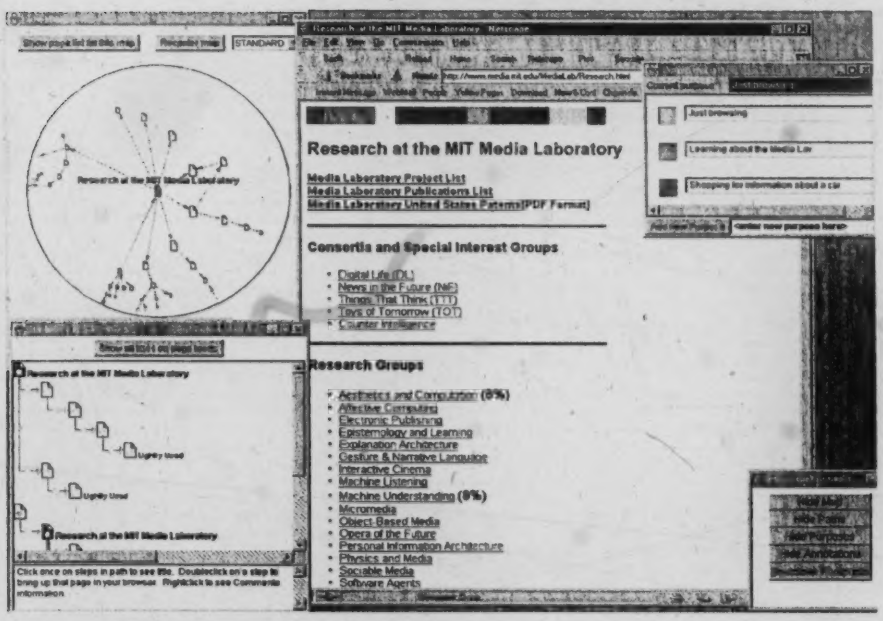


Figure 2.27: FootPrint

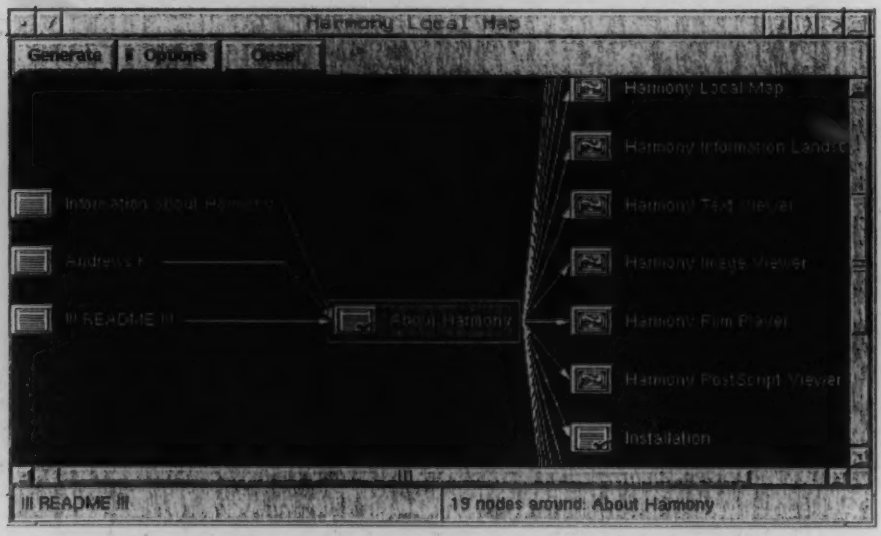


Figure 2.28: Harmony Internet Explorer

could without a map.

In this chapter, Tutfe's visualization design principles are enlisted first, some examples are given to illustrate them. Second, different web site structure modelling methods are discussed. Third, pros and cons of the currently available web site structure visualization systems are evaluated. Finally, the web mining and web navigation visualization systems are reviewed in details. In Chapter 3, the WebKIV system architecture will be introduced. The data pre-treatment and transformation are discussed first. Then radial algorithm to map the web site structure is detailed. The various data operators of WebKIV system are covered next. At last the functions to observe the data visualized in WebKIV are explained.

Chapter 3

WebKIV System Architecture

The previous chapter presented a literature review. This chapter will introduce the Web Knowledge and Information Visualization (WebKIV) System. Some of the visualization design principles will be discussed in Section 3.1. The system architecture and data flow will be detailed in Section 3.2. Section 3.3 will discuss the radial tree algorithm. The data manipulating interface will be covered in Section 3.4. Finally, Section 3.5 will present the functionality of the WebKIV System.

3.1 Visualization Principles Discussion

In Section 2.1, several visualization design principles were discussed. These principles are satisfied by the WebKIV system design process.

Multiple Views. The WebKIV system designed for this present dissertation adopts a multiple views feature. While one view window (the Context window) always shows the context information of the whole web site structure, the other view window (the Focus window) displays the detail of what the user chooses to display, a subset of the useful information determined by the user (Figure 3.5). The context switching problem is alleviated by correlating the two view windows. Whenever there is a need to drill down, the subareas chosen for zooming are rectangled in the context window. The zoom effect is shown in the focus window in detail. As

drilling down continues, the changes will be reflected in the context window by changing the rectangle accordingly.

Interactive Interface Design. In the design of the control panel, the information is stratified and separated to give the user flexibility to decide what to isolate, what to emphasize, etc. Threshold settings and animation speed settings are implemented to adapt the visualization system to different users and different data sets.

Small Multiples. This idea is also implemented in the WebKIV system. For a specified area, topologically identical graphs can be generated in order to show the differences among the different data sets.

Colour Settings. According to the "Colour and Information" principle of Tufte, the colours in the present system are chosen very carefully. All the colours chosen are the normal ones that can be easily found in nature. Bright colours are used only sparsely, in order to reveal the useful patterns. Thick lines are used to emphasize what is important, although they can obscure each other. A detailed explanation is presented in Section 3.4.

3.2 WebKIV System Architecture

3.2.1 Data Collection Phase

Three kinds of web data are collected to be visualized.

- Web Site Topology Data.

A web crawler is used to get the individual web site structure information. The hyperlinks and URLs of web pages are recorded. The record contains either only HTML web pages information or the information of all the web pages including .ps, .gif and .wav.[1] A more detailed algorithm description is presented in Section 3.3

[1]The web site structure data used in this thesis contains only HTML pages.

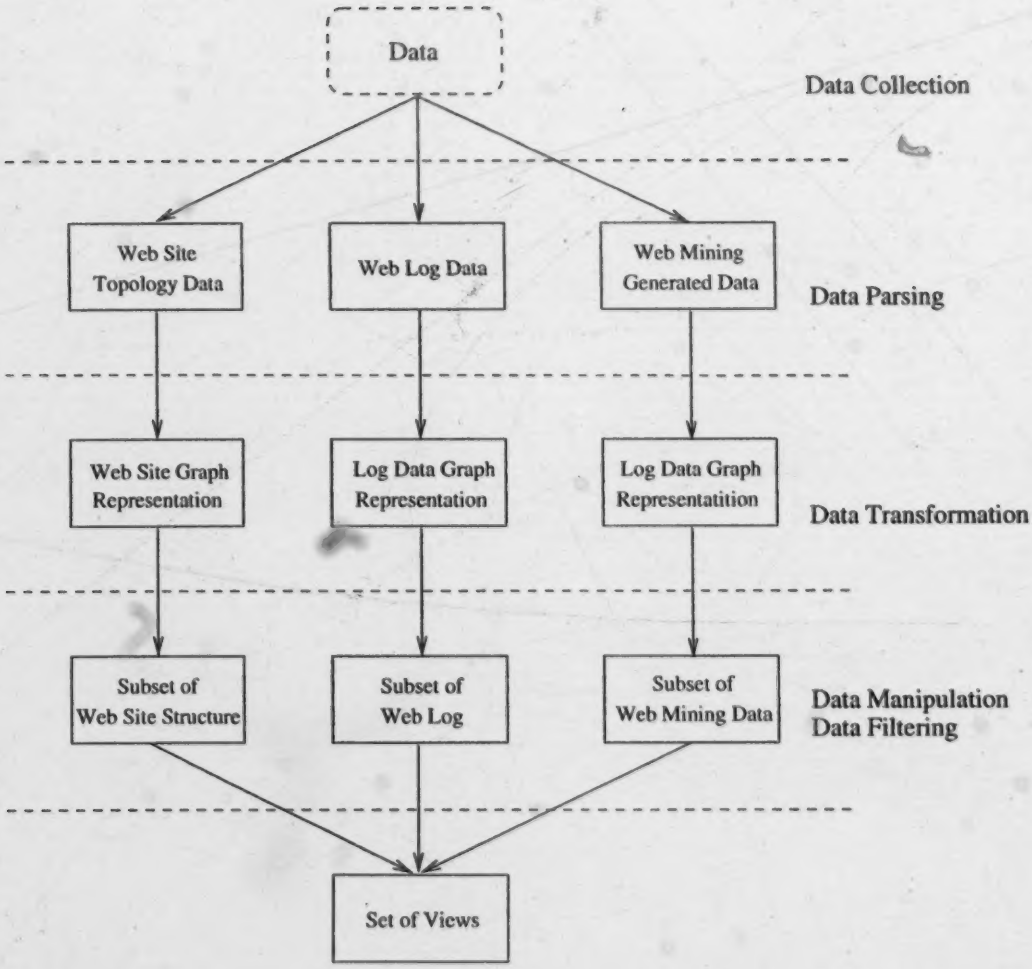


Figure 3.1: WebKIV System Architecture

- Web Log Data

A web log file is a text file in which every page requested from the web server by a user is recorded. The access patterns can be stored in 2 different formats: the Common Log Format(CLF)[2] and the Extended Common Log Format(ECLF). Each record will contain the following fields:

- IP address or domain name of the computer which sends a request to the web server
- UserID and password
- Date and Time of the request made
- Request. Includes request type like GET, POST and HEAD, the documents the request needs and the protocol (FTP or HTTP, for example)
- Status Field. Indicates the status of the request; for example, it indicates if the request was successful
- Transfer Volume. Indicates the number of bytes transferred to the user.
- Referrer. Tells where the user comes from . "-" is used in this field for "no referrer".
- Name and version of the user browser.
- Name and version of the user operating system.
- Cookie. A unique user identification generated by the web server for each user.

[2]CLF does not have a Referrer and an User browser and operation system information.

Below is an example of log entry from the access log of the Department of Computing Science at University of Alberta

> *123.12.13.1 - - [01/Aug/2001:00:01:44 -0600] "GET /~chinook/people/authors.html HTTP/1.0" 200 4424 "http://web.cs.ualberta.ca/~chinook/" "Mozilla/4.7 [de] (WinNT; U)" "uabcs=213.162.65.12.2149999664563662"*

Before the raw web log data can be subjected to any information extraction methods, data filtering, user identification and session identification must be applied first [20]. The web log file is not designed to be used by web mining, but to record everything the server does, in order to help the web administrator to monitor the web server and find any problems. Before the web log can be used, the unsuccessful requests and the requests not related to the HTML files such as gif and jpg files must be dropped off[3]. Then the log entries for each user will be grouped respectively. Finally, the click stream of a web surfer during a single visit (session) to a web site is obtained.

- Web Mining Generated Data.

A simulated web log generated by web mining techniques belongs to this category. Although Association Rules, page clusterings and user clusterings are typical web mining results, they are not visualized directly in the WebKIV system. Instead, by applying these web mining results, new web log data can be generated and compared with the original.

In the present thesis, the web log files generated by Navigational Compression Models (NCMs) [60] are used in the system to evaluate the effectiveness of the web mining results. The basic principle of NCMs is to discover navigation patterns

[3]in some web sites, image files can be the important information which should be visualized.

from previous visitations, and then to use these patterns to provide guidance for new users in order to save them from taking "unnecessary" steps.

By visualizing the difference between the log data before applying NCMs and after applying the learning methods, the WebKIV system can show the "saved" clicks.

3.2.2 Data Parsing Phase

In the data parsing phase, web log, web site structure and contents, and web mining data will be parsed and filtered first before they can be displayed in WebKIV.

Web site hierarchy. Web site hierarchical structure is tightly coupled with the radial tree algorithm. The parent-son relationship is reserved while the radial tree algorithm is applied to the web site. As well, the URLs as the content representation of the web pages are also saved. However, There is a mismatch between the web site structure and the web logs. Web sites keep changing: some web pages are added to the web site at one time, whereas some are deleted at another time. Thus, the web site structure and parent-son relationship constructed today is only good for today's log data. To resolve this problem, two approaches are available. One is to construct the web site based on all the web pages which were in the web site during a period of time [15]. The benefit to keeping all the web pages is that all the web log information are kept and displayed without a loss. The disadvantage, however, is that the operation of "subtraction" of the web logs (see Section 3.5.5) might generate distracting results. For example, it could be that a small group of web pages are shown visited extensively in WebKIV under the subtraction operator only because these web pages are newly added. In WebKIV, we take a simpler approach. A web log record is dropped if the web page visited is not in the web site structure data. Although the latter solution will lose some of the web log records, we assume that the loss is very likely to happen at the bottom of the hierarchical

tree and the data loss is small.

Session data. In WebKIV, all the statistics and patterns are retrieved directly from session data files. The reason that sessions are used instead of the raw web log records is that the noisy data, such as the non-successful requests, spider programs' traffic which will skew the statistic results, etc, have already been filtered out[4]. However, users of WebKIV have the freedom to use the raw web log data as well. An example of a session used in WebKIV is shown in Table 3.1.

Statistics of web page visits. The algorithm to get the web page visits count is straightforward: for all the sessions, count the visits for each web page. An example of the data format is shown in Table 3.2. The web pages logged in web logs but not found in web site structure data will be dropped.

Statistics of user traversal pattern. The user traversal pattern data is also derived from session files, as is shown in Table 3.3. In WebKIV, one hyperlink is treated as one traversed unit and the counts are calculated. Since the lines used in WebKIV are straight, the stroke overlapping such as A -> B and B -> A is inevitable. However, this problem can be alleviated. The traversal counts are transformed to the thickness and colour of the lines. Although a line with light colour might cover the line with dark colour because of the order of drawing them, the thickness of the line can still tell the traffic of the hyperlink which has a higher traffic volume .

Subtraction of two web data sets. Currently in WekKIV, data set subtraction applies to two kinds of data: statistics of user traversal pattern and statistics of web page visits count. For instance, the subtraction of web page usage count between different time. (Figure 4.7)

Web user surfing animation. Session data is used for web user surfing animation. The difference between web user surfing animation data set and session data set is the order of the rows. Web user surfing animation data set is ordered by the time

[4]How to get a session is beyond the scope of this thesis, interested users please refer to [20]

only (Table 3.4).

3.2.3 Data Transformation

In WebKIV, three shapes are used to represent the web data.

Node representation. Each web page is represented by a square. The square has no colour representation to bear because it is often too small to make a colour difference.

lines representation. In WebKIV, a stroke has two attributes: width and colour. While strokes represent different data sets, width and colour are used to display the characteristics of the data sets.

In the context of *web site hierarchy structure*, a parent-son relationship is represented by a line connecting two web pages. For *web page visits count*, since the squares which represent the web pages are too small to bear any information, lines are used to represent the count of web pages being accessed. Each line connecting the parent and son represents the visit counts of the son. The darker the colour, the wider the line, the more the web pages are visited (Figure 4.3). In the context of *web traversal pattern,* the lines connecting any of the two nodes represent the real hyperlinks. The width and the colour of the lines represent the traversal patterns (Figure 4.5).

While for most of the WebKIV data sets, lines are good enough to display the various web activities, the representation of *user surfing animation* is a little different. A *dot* is used to represent a web surfer. The dot will move during the animation process, representing the user is moving from one web page to another through a hyperlink. Lines will still be used to show the traversal patterns. The difference is that the colour of the lines will change over time based on the traffic (Figure 4.10).

Session ID	User ID	Time Stamp	URL
1	anon0149958	01/Jan/1999:23:17:13:610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
1	anon0149958	01/Jan/1999:23:19:7:252	machines.hyperreal.org/Analogue-Heaven/mail/ah.9807.html
1	anon0149958	01/Jan/1999:23:19:20:610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
1	anon0149958	01/Jan/1999:23:19:27:306	machines.hyperreal.org/schematics.html?MMAgent
1	anon0149958	01/Jan/1999:23:19:41:610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
2	anon0149819	01/Jan/1999:23:17:47:28	machines.hyperreal.org/samples.html?MMAgent
2	anon0149819	01/Jan/1999:23:18:4:12	machines.hyperreal.org/
2	anon0149819	01/Jan/1999:23:18:11:10	machines.hyperreal.org/links/
2	anon0149819	01/Jan/1999:23:18:24:12	machines.hyperreal.org/

Table 3.1: Examples of Session format

URL	Assigned ID	Visits Count
machines.hyperreal.org /manufacturers/Rhodes/Chroma/mods/	1433	18
machines.hyperreal.org /manufacturers/Opus/	670	81
machines.hyperreal.org /manufacturers/Roland/CompuRhythm/mods/	1436	2
machines.hyperreal.org /manufacturers/Yamaha/RX/info/	668	27
machines.hyperreal.org /categories/monosynths/Pro-One/patches/	661	117

Table 3.2: Example of web page visits count

Assigned ID	Traversal Count
machines.hyperreal.org/manufacturers/EVS/ -> machines.hyperreal.org/manufacturers/EVS/info/	28
machines.hyperreal.org/samples.html -> machines.hyperreal.org/categories/drum-machines/samples/	7
machines.hyperreal.org/manufacturers/Opus/ -> machines.hyperreal.org/manufacturers/Opus/Opus/	16
machines.hyperreal.org/samples.html -> machines.hyperreal.org/manufacturers/Korg/KR-55/samples/	4

Table 3.3: User traversal pattern count

3.2.4 Data Filtering & Manipulation

Rotation, zooming and panning, subtree generation, threshold setting, and animation display can be done by users to interactively observe the web data.

User ID	Time Stamp	URL
anon0149958	19991123.19.41.610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
anon0149958	19991123.19.27.306	machines.hyperreal.org/schematics.html?MMAgent
anon0149950	19991123.19.25.311	machines.hyperreal.org/manufacturers/Roland/MC-202/
anon0149961	19991123.19.20.199	machines.hyperreal.org/manufacturers/Moog/
anon0149958	19991123.19.20.610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
anon0149960	19991123.19.14.12	machines.hyperreal.org/
anon0149958	19991123.19.7.252	machines.hyperreal.org/Analogue-Heaven/mail/ah.9807.html
anon0149959	19991123.18.42.79	machines.hyperreal.org/manufacturers/Encore/info/
anon0149819	19991123.18.24.12	machines.hyperreal.org/
anon0149819	19991123.18.11.10	machines.hyperreal.org/links/
anon0149956	19991123.18.10.13	machines.hyperreal.org/adaptive/2.html
anon0149819	19991123.18.4.12	machines.hyperreal.org/
anon0149956	19991123.18.0.13	machines.hyperreal.org/adaptive/2.html
anon0149819	19991123.17.47.28	machines.hyperreal.org/samples.html?MMAgent
anon0149956	19991123.17.18.13	machines.hyperreal.org/adaptive/2.html
anon0149958	19991123.17.13.610	machines.hyperreal.org/manufacturers/Korg/MS-synths/schematics/service-manual/
anon0149957	19991123.17.5.664	machines.hyperreal.org/categories/samplers/ESI-32/info/
anon0149956	19991123.16.7.13	machines.hyperreal.org/adaptive/2.html
anon0149956	19991123.15.45.12	machines.hyperreal.org/

Table 3.4: User surfing animation example

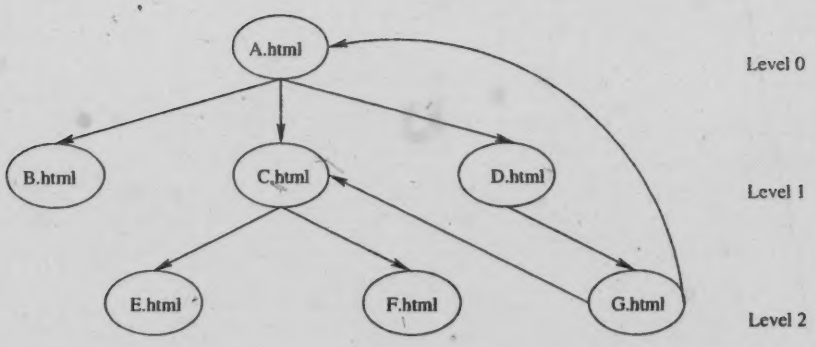


Figure 3.2: Web site structure

3.3 Web Site Mapping Algorithm

The radial tree is a hierarchical acyclic tree. All the nodes are assigned to different concentric circles according to their levels in the hierarchical structure of the web tree. A web site structure, however, is not a hierarchical structure, but a cyclic directed graph. Determining the parent - son relationship between web pages, which is a must for a hierarchical tree, is difficult. The method we use is to use a breadth-first search to walk within a web site and to build the hierarchical structure according to the order in which the walker or web crawler walks. Only the links supporting the structure are saved, and rest of the links will be cut off. The radial tree algorithm adopts the same method to get the web site structure. The following example illustrates how a web site structure becomes a hierarchical tree via the breadth-first search web walker:

The Figure 3.2 shows a mini web site with 7 web pages. The lines represent the hyperlinks, and the arrows correspond to the direction of the link, e.g., from G.html

Node	Father	Son	Leaf Node Number	Level
A	–	B,F,G	20	0
B	A	C,D,E	10	1
C	B	Leaf	5	2
D	B	Leaf	2	2
E	B	Leaf	3	2

Table 3.6: Radial tree Parent-Son relationship

to A.html. A web walker usually starts from the homepage, which in the example is A.html. According to the breadth-first traversal algorithm, A.html is the first page the web walker explored. A.html is the root of the tree and records to level 0. A.html has 3 hyperlinks, from left to right, B.html, C.html and D.html, respectively. B, C and D are the sons of A.html, and their level in the tree is 1. B.html does not have any out links, so B.html is also a leaf node. C.html has two out links, named E.html and F.html, so E.html and F.html are sons of C.html, and their level is 2. D.html has one out link to G.html, so G.html is the son of D.html, and its level is 2. Since no nodes are in level 1, the crawler continue to process the nodes in level 2. E.html and F.html are leaf nodes. G.html is also a leaf node. Although G.html has two out links to A.html and C.html, A.html and C.html are not sons of node G, because A.html has already been visited by the web walker and been assigned to the tree, so A.html cannot be the son of G.html. For the same reason, C.html is also not the son of G.html. By cutting off the circle according to the visit sequence, the acyclic hierarchical tree structure shown in Figure 3.3 has been constructed.

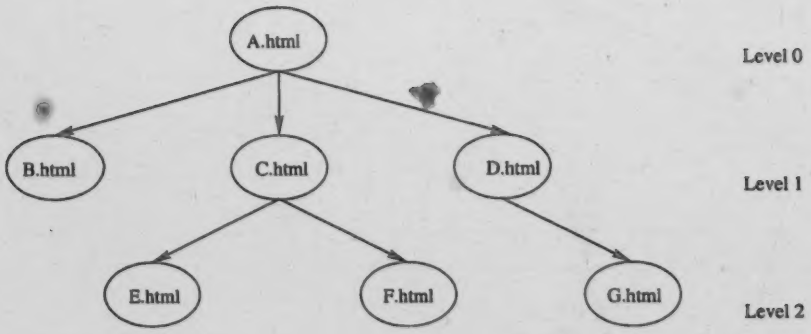


Figure 3.3: Web site tree structure

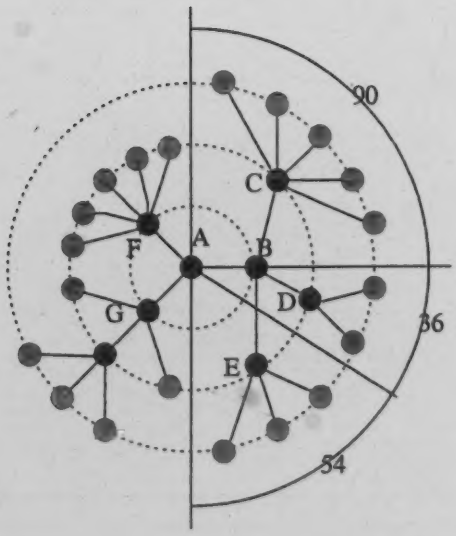


Figure 3.4: Radial tree structure

After the web site tree structure has been constructed, the radial tree algorithm can be applied to it. The radial tree algorithm traverses the web site tree twice, the first time in a post-order sequence to calculate the number of leaf nodes for each subtree. For each node, the post-order traversal calculates the number of leaf nodes (leaf nodes are the nodes which do not have any son nodes) that are rooted from that node. The first traversal provides the total number of leaf nodes so that the angle span for each leaf node can be calculated:

one leaf node angle span = 360/total number of leaf nodes in the tree.

Second, the breadth-first search traversal algorithm is used to recursively assign angles for each subtree:

subtree node angles = one leaf node angle * number of leaf nodes rooted from this node.

In Figure 3.4, node A is the root node. It has 3 sons and 20 leaf nodes in total. (See Table 3.6 for the hierarchy and leaf number of each node.) Node B has 10 leaf nodes. Node C has 5 leaf nodes, and so on. According to Disktree definition,

one node angle = 360/20 = 18 degrees.

so one leaf node will have 18 degrees.

for Node B, the subtree leaf node angle = 18* 10 = 180 degrees.

for Node C, the subtree leaf node angle = 18*5 = 90 degrees.

for Node D, the subtree leaf node angle = 18*2 = 36 degrees.

for Node E, the subtree leaf node angle = 18*3 = 54 degrees.

.

.

.

This process continues until all the nodes have been positioned.

3.4 Data Operator Details

A visualization designer often has difficulty knowing exactly what the user wants to see. Besides, the abstract data is often multidimensional, so showing all the attributes of the data in one view is typically not possible nor necessary. The layering and separation principle has to be used to prevent users from becoming confused with information overload. Good flexible interaction design is required to satisfy both the different user requirement and different web site data sets. In the WebKIV system several general interaction methods have been implemented.

Multiple views. The WebKIV system has two display windows: the Context window is used for the context overview, the Focus window is used for drill down. A rectangle in Context window shows which part of the web site is displayed in the Focus window.

Zooming and Panning. Zooming and panning operations are provided in order to allow users to explore and switch from part of the information space and the whole data set. Zooming provides users not only with larger-sized shapes, but also with detailed information such as the web page URL, which cannot be viewed in a Context view.

Two zooming operations are provided in the WebKIV system: a rectangled zoom in an area and a zoom in a subtree of the radial tree. A zoom in a rectangled area is made via a rectangle which can be drawn in either the Focus or Context window. The nodes and lines in the rectangle will be redrawn in Focus window (Figure 3.5). The ratio of the scaling depends on the ratio of the height and width of the rectangle and that of the Focus window. After the scale rate is decided, the center of the rectangled area is shifted to the center of the Focus window to keep the enlarged graph in the middle of the window; then the positions of the nodes are recomputed, and the nodes and lines are redrawn. To synchronize the two windows, if the rectangle was drawn in the Focus window, it is drawn in the Context window

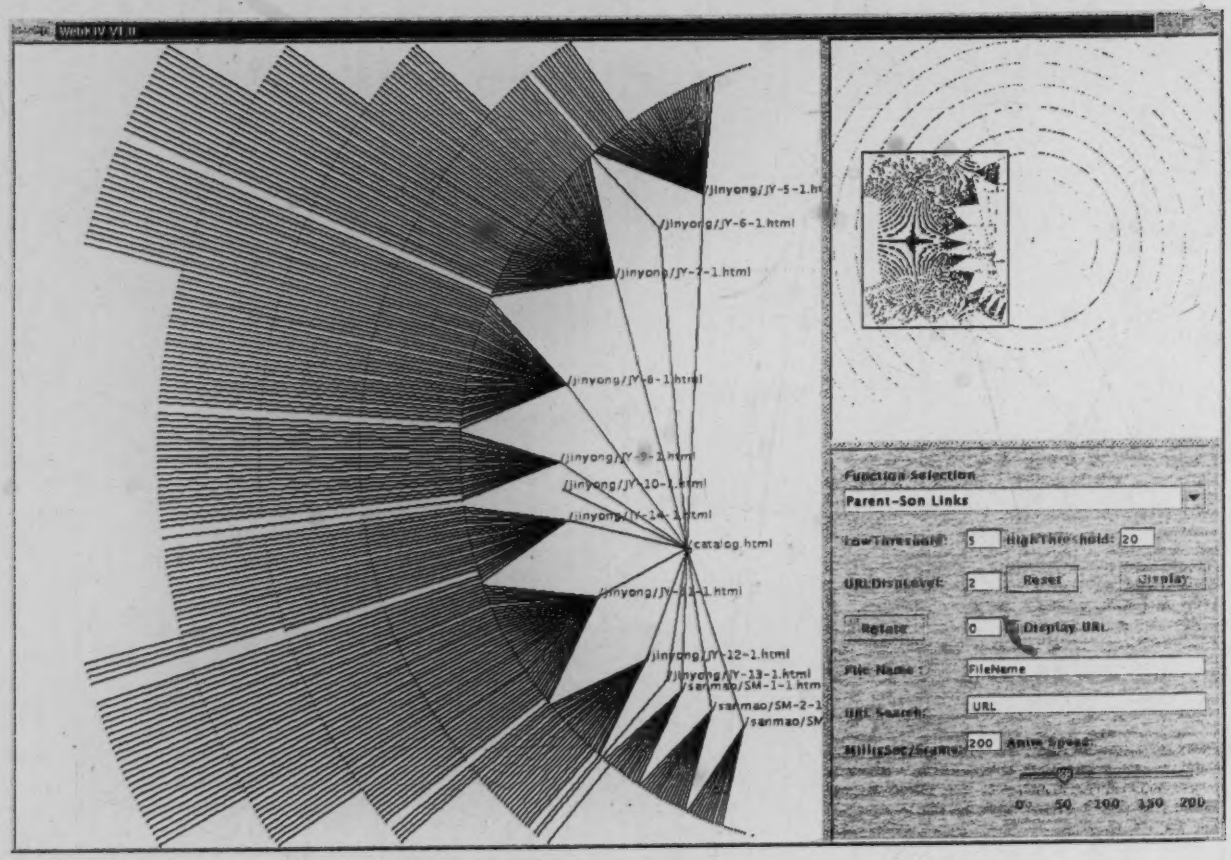
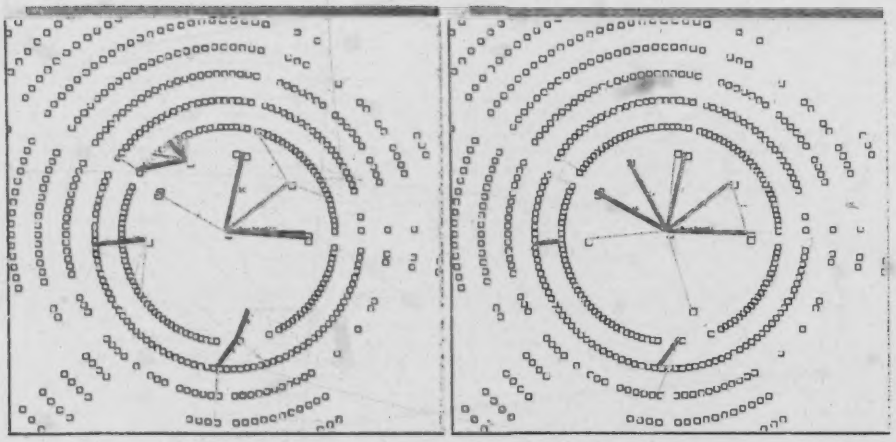


Figure 3.5: WebKIV System
Left: Focus View window **Right:** Context window



(a) (b)

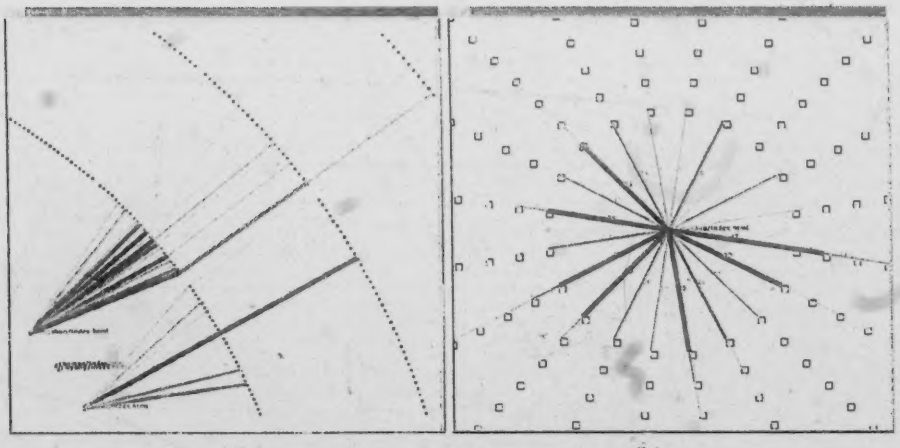
Figure 3.6: Identically generated subtrees

(a): Web page usage representation **(b)**: Hyperlink usage representation

to show the part of the graph that was enlarged.

The "Subtree Generator" is the other zoom-in technique used to provide a detailed view of a subset of a web site. The differences between the rectangled zoom-in and subtree zoom-in are the following (Figure 3.7): First, the rectangled zoom-in enlarges and displays the information in the rectangle, whereas the subtree zoom-in enlarges and displays only the information related to the nodes that are rooted from the node selected. Second, the deployment of the nodes also differs. The rectangled zoom-in does not change the pattern when enlarged, whereas the subtree zoom-in generates a new radial tree centered with the node chosen to be the root node. The subtree generation operator can also be used to generate identical web maps so that different data attributes can be displayed side by side at same time without overlapping each other (Figure 3.6). This feature makes data comparison and data correlation very easy.

The advantage of the rectangle zoom-in is that it keeps the pattern as it was, but the space is not used as efficiently as it is by the subtree zoom-in, as is shown



(a) (b)

Figure 3.7: Rectangled Zoom-in and Subtree Zoom-in
(a) Rectangled zoom-in (b) Subtree zoom-in

in Figure 3.7 (a). The subtree zoom-in generates a radial tree and puts the most important patterns in the center of the view. The problem with subtree zooming is that it completely changes the patterns, users might feel confused (Figure 3.7 (b)).

Threshold Setting. Threshold setting is very important for making any differences "stand out" from the background and for eliminating the noise data from views.

At the beginning of the WebKIV design phase, only colours were used to show the data attributes. For example, the more a hyperlink being used, the brighter the colour of the line representing the hyperlink. Initially, the thickness of lines was not used to represent the data attributes. The concern was that the thick lines would block out their "neighbouring" thinner and dimmer lines, but tests showed that the important parts (the most traversed lines and the most visited nodes) were not emphasized enough. Therefore, both line thickness and colours are used to highlight the most important information in the present WebKIV system.

Another design issue related to the threshold setting involves how to distribute

Figure 3.8: Pan view of web page visits statistics

the colour and the thickness of lines in relation to different threshold settings. In the WebKIV prototype, only the data which fall in between the lower threshold and higher threshold can be displayed. The rest of the data is not displayed. The benefit of the segmentation is that it is more focused on the region data which the user has selected. The problem, however, is that users hardly know the upper limits of the data set. They have to set a very high number as the upper limits in order to include the most important patterns. This drawback causes the following problem: the number of colours which the users can clearly distinguish is below 30 [55] and the range between the thresholds is too big to distribute the colour. For the thickness of a stroke, 14 times is the biggest difference range in the present system; thicker than that will block out the other useful information. In the current WebKIV system, both the lower bound and upper limits are set, but all the information above the lower limits will be displayed. The upper limits are used only for setting a relatively small range to distribute the colour and thickness of lines.

URL Display. Two URL display operators are provided in the WebKIV System. A "LeveltoDispURL" operator is used to display URLs according to the node level a node sets in the radial web tree. A "Display URL" checkbox is a display switcher which can be set to either display all the nodes URLs or display none URLs.

Animation Speed Setting. Two operators are related to the user navigation animation function. One is the "FrameRate" setting. It tells the WebKIV system how many times per second the view should be refreshed. The faster the frame refresh speed, the smoother the animation. Usually the frame refresh rate should not be less than 5 frames per second in order to keep the animation fluid. The frame refresh rate and system performance will be discussed in detail in Chapter 5. Animation Speed is another operator for deciding how many times faster than real-time the log should be processed. A slider is provided for users to choose the times from 1 to 200. No universal rules applicable to determine what ratio is suitable for a web site. The ratio varies greatly according to the traffic and the size of the web site. From [42], 50 times real-time is an appropriate setting for a moderate traffic web site.[5]

Rotation. In display of the nodes' URLs, the URLs of nodes positioned at the top and bottom of the circles will overlap each other, thus making the URLs unreadable (Figure 3.9). To solve the problem, which is inherent in a radial tree, rotation is provided to make the nodes move from the top and bottom to the left and right sides.

Node Position Setting. The WebKIV system allows viewers to change the settings for the levels on which the nodes are displayed. Although displaying all the nodes on the screen seems easy and necessary, problems result when one of the subtrees is too deep. For example, in Figure 3.10, the node level display is set to maximum. Since some subtrees are very deep, most of the nodes are positioned very densely. The valuable screen space is not used efficiently. To avoid this prob-

[5]50 times works well for the traffic of web site http://machines.hyperreal.org. The web log files are from http://www.cs.washington.edu/ai/adaptive-data

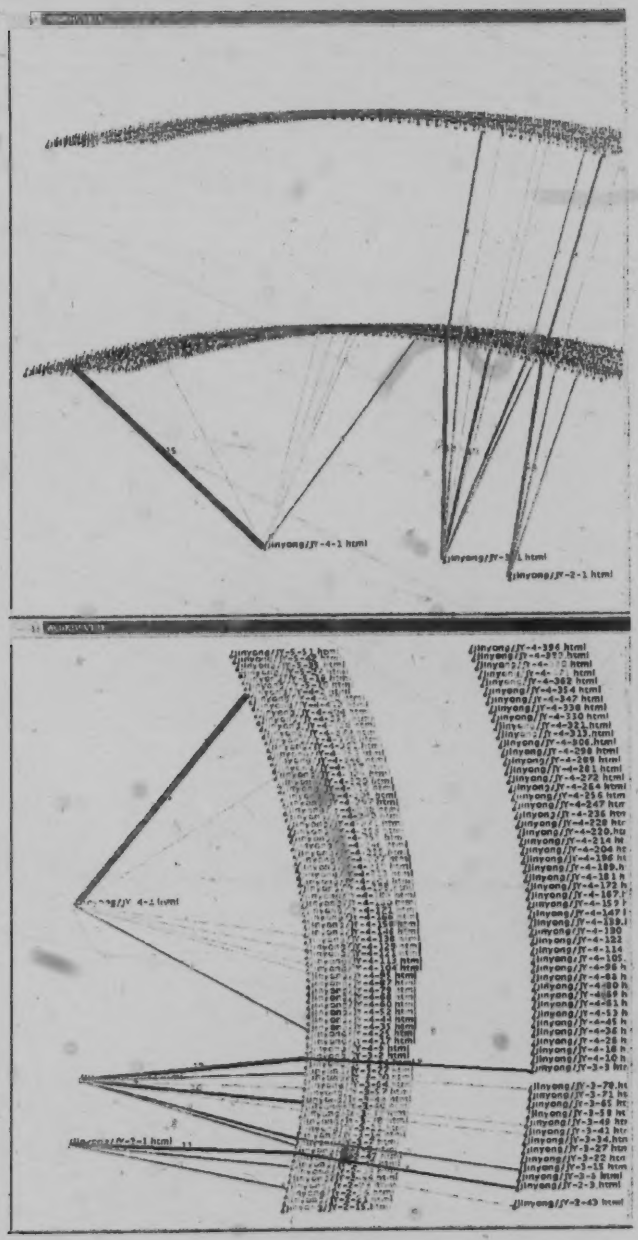


Figure 3.9: Rotation example
Top: before rotation **Bottom:** after rotation

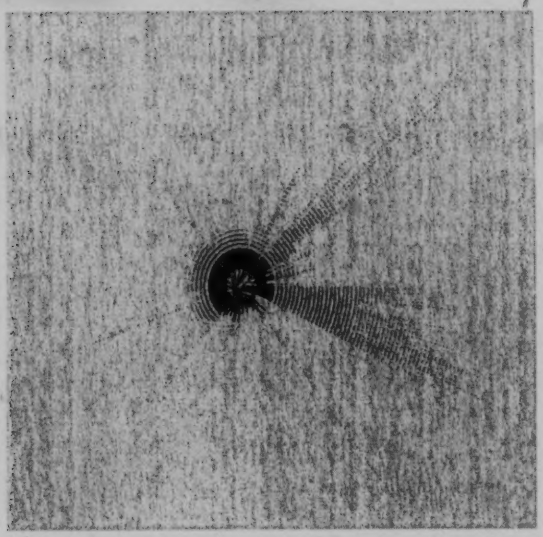


Figure 3.10: Maximum node level setting

lem, viewers of WebKIV can set a number to indicate how many levels of nodes to display.

3.5 WebKIV Functionality

Several ways to display web attributes have been implemented in the present WebKIV System. They are:

Web Site Structure,

Web Site Hierarchy,

Statistics of Web Page Visits,

Static User Traversal Pattern Display,

Subtraction of Two Web Data Sets and

Web User Behaviour Animation.

While each function works well on its own, the structure of the data sets can be better revealed when observed and manipulated by applying the functions together

(please refer to Section 4.2.1.3 for an example).

3.5.1 Web Site Structure

It displays the initial view of a web site generated by radial tree algorithm. Viewers can get back to this view from any data manipulation phase at any time by pressing "Reset" button. All the data sets will be re-initialized.

3.5.2 Web Site Hierarchy

The lines will be drawn from "parent" nodes to "son" nodes so that the hierarchy relationship can be displayed (Figure 4.2). The web site structure visualization has many potential applications on its own. For example, if a branch (a subsection of radial tree) is very deep compared with other branches, it may indicate that some of the information is too far away from the home page. It may further imply that the user might take more steps to reach the pages. The web site hierarchy can also directly reveal the web site "balance", e.g., the span of one branch is too wide and the others have only few descendants.

3.5.3 Statistics of Web Page Visits

This function will show the web page visits frequency (Figure 4.3). Lines drawn from "parent" nodes to "son" nodes represent the count of visits of the "son" nodes. The colour and thickness of the lines indicate the number of times the web page was visited during a fixed period of time. The threshold can be set to segment the lines being displayed. The threshold can also be adjusted to make part of the lines thicker and brighter according to viewers preference. This way, the "hot spots" of a web site are easy to be identify.

3.5.4 Static User Traversal Pattern Display

In this function, the hyperlinks the user traversed while browsing a web site are shown (Figure 4.5). It shows the hyperlinks the user used while browsing a web site. It reveals the correlation of the web pages and the web users traversal behaviour. For example, it can show which pages are visited together and which hyperlinks the users often used.

When used with the Function "Statistics of Web Page Visits", it is easy to answer questions such as " From where the users arrived at the most visited web pages?" or " Are there some links that contribute large number of visits to a 'hot spot' page?" (Figure 3.6)

3.5.5 Subtraction of Two Web Data Sets

Subtraction is used to show the difference of the two web data sets. It is useful if the difference of the two data sets is small. With the subtraction of the two data sets, the common part of the data is deleted from the view, only the difference is shown. The data sets can be collected from different period of time to show the trend of the web usage over different time periods; or they can be generated by various web mining knowledge to compare the effectiveness and the efficiency of the different web mining methods.

3.5.6 Web User Behaviour Animation

Dynamic web user navigation visualization applies the idea in [42] (Figure 2.18). The difference is that the idea was generalized in WebKIV to apply to automatic generated web site structures instead of the hand made web site structure. Another advantage of the WebKIV system is that , with appropriate path annotation, we can dynamically observe aggregate behaviour, e.g., aggregate navigation path changes over different time periods (Figure 4.10).

Chapter 4

Experiments

Effectiveness of pattern finding, interactivity of data manipulating, and scalability for large data sets are the three key features to test in experiments.

4.1 Data Preparation

4.1.1 Data Source

Two web sites and their web data sets were used to test the WebKIV system.

`http://www.cs.ualberta.ca`

This is the web site of the Department of Computing Science of the University of Alberta (UACS).

This web site contains about 58989 html web pages. A one-month web log file (Oct , 2001) was used for the experiments. The data are shown in Table 4.2.

`http://machines.hyperreal.org`

This is the web site of Music Machines, which contains 1304 html web pages. A two-month web log file (Jan, March, 1999) was used to test the web site structure. The data are shown in Table 4.1

4.1.2 Data Format

The data are shown in Table 4.2 and Table 4.1. The explanation of the table's content is listed below[1]:

Total number of log entries: The total number of requests that surfers requested and delivered successfully, no matter what type the files are.

Total number of HTML logs: The total number of HTML web page requests recorded by the web server, not including the image, compressed file and audio file, etc.

Total number of users : The total number of surfers.

Total number of sessions: The number of surfer's click stream segmented by the time interval of consecutive HTML web page clicks.

Total number of sessions (length > 1): The number of sessions containing at least two HTML web page clicks.

The data used in the WebKIV system is from the sessions whose length is longer than one.

4.1.3 Navigational Compression Models (NCMs) Data

Navigation Compression Models (NCMs) is an application based on the knowledge mined by web usage mining algorithms. It has three parts: Knowledge Base, Recommendation System and Feedback System. The Knowledge Base contains knowledge mined by clustering, association rule, etc. The Recommendation System gives users suggestions based on user behaviour after entering a web site.

NCMs provides a way to observe the effect of web site redesign. One problem with web mining application is that although various mining methods provide some information patterns, we do not know if they are useful; that is, we do not know

[1]The data parsing such as user identification and session identification is beyond the scope of this dissertation, interested viewers please refer to [19]

	Total No. of Logs	Total No. of HTML logs	Total Users Identified	Total Sessions Identified	Total Sessions Length > 1	Total Sessions Length = 1
Jan, 1999	795833	202663	33858	53301	27728	25572
Mar, 1999	880654	217322	33352	55313	27115	28190
Jan,week 1	180923	46299	8398	11700	6133	5567
Jan, week 2	187002	47939	9023	12660	6645	6015
Jan, week 3	192370	47965	9093	12644	6611	6033
Jan, Week 4	185209	48243	9166	12825	6606	6218

Table 4.1: Music Machine Web Log File for Jan/99 and Mar/99

	Total No. of Logs	Total No. of HTML logs	Total Users Identified	Total Sessions Identified	Total Sessions Length > 1	Total Sessions Length = 1
Sep, 2001	3258857	514343	55753	93419	74918	18491
Oct/01-07/01	698875	100893	13958	18532	14851	3680
Oct/08-14/01	755257	120830	14652	21261	16949	4309

Table 4.2: UACS Web Log Data

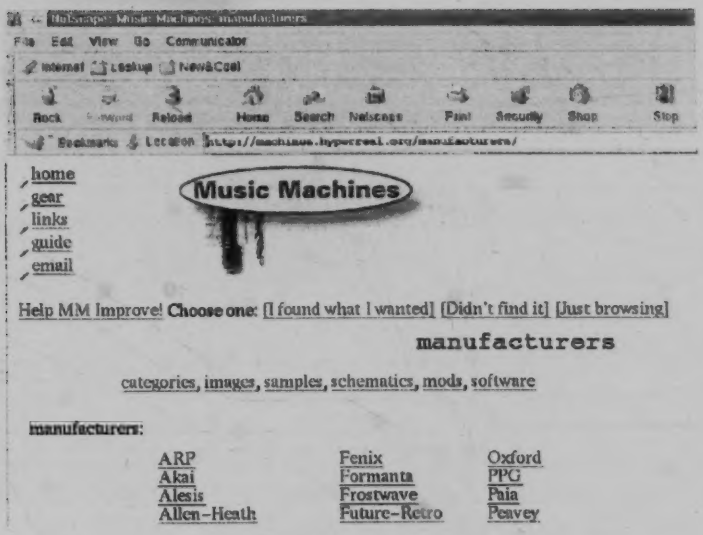


Figure 4.1: Music Machine web page snapshot

how to evaluate the discovered patterns. One option is to modify the web site based on the knowledge mined and collect user feedback or analyze the new web log generated to evaluate the efficiency of the web site redesign. While doing so is the most accurate way, an often changed web site will cause user confusion and impatience.

Instead of changing the real web site, NCMs can simulate the user behaviours on a modified web site. NCMs can generate new web log entries based on the original user web log. Here we assume the users have the same goal in mind and that the recommendation system does not affect user behaviour.

4.2 Experiments

4.2.1 http://machines.hyperreal.org

4.2.1.1 Web Site Structure

This web site introduces musical instruments. It provides images, an introduction and samples of and comments on various electronical equipment such as synthesizers and drum machines.

The structure of the web site is shown in Figure 4.2, The upper right part of the web site, which is rooted from `http://machines.hyperreal.org/categories` has a different organization compared to the lower left part, which is rooted from `http://machines.hyperreal.org/manufacturers`. The subtree of "categories" has a deep and narrow tree structure, and the subtree of "manufacturers" has a shallow and wide tree structure. Note that both parts provide identical information; e.g., each file (text, image, wav files, etc., not the html files) in one part can be found in another part of the web site.

4.2.1.2 User Visits Distribution

Figure 4.3 reveals that the subtree of the web site rooted from `http://machines.hyperreal.org/manufacturers/` is well used (left part of graph), whereas the upper right part rooted from `http://machines.hyperreal.org/categories/` and the upper left part rooted from `http://machines.hyperreal.org/images/` are seldom visited.

As we zoom in on the well visited subsection rooted from the manufacturers, four well-visited web pages are shown in Figure 4.4:

1. `http://machines.hyperreal.org/manufacturers/ARP/Odyssey/samples/`;
2. `http://machines.hyperreal.org/manufacturers/Moog/MiniMoog/samples/`;
3. `http://machines.hyperreal.org/manufacturers/Moog/MiniMoog/samples/` and
4. `http://machines.hyperreal.org/categories/drum-machines/samples`.

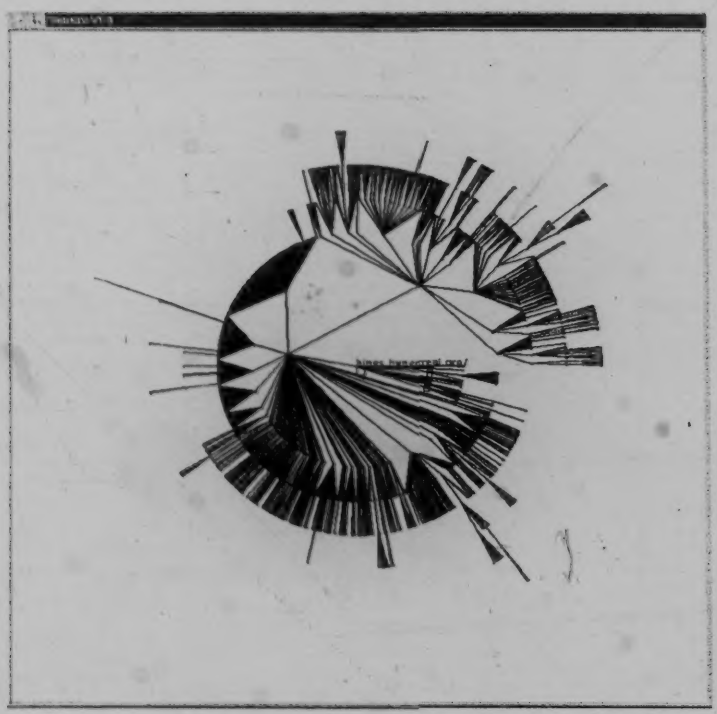


Figure 4.2: Music Machine Web Site Structure

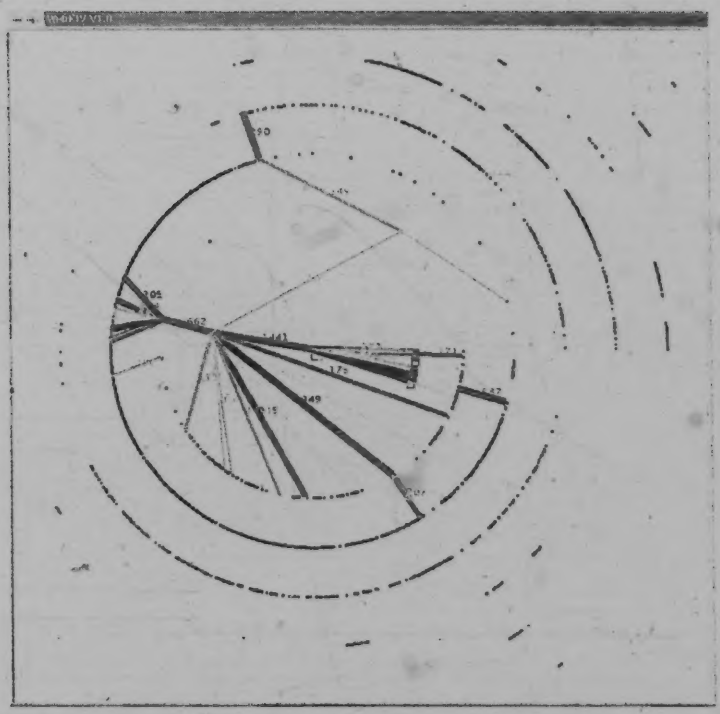


Figure 4.3: Music Machine visits distribution.
Visits frequency threshold is set between 50-450.

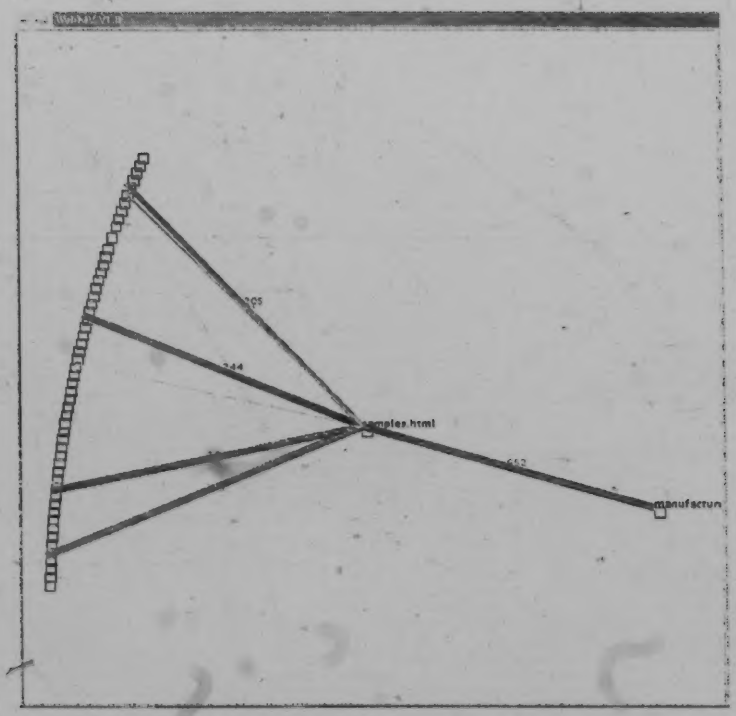


Figure 4.4: Music Machine Rectangled Zooming
The well visited leaf nodes

The well-visited graph implies that the users can most easily find the specific musical instrument sample web pages on this part of the web site. For example, the web page `http://machines.hyperreal.org/manufacturers/Roland/TR-909/samples` has an identical web page copy which can be found via another hierarchy route: home page -> categories -> drum-machines -> TR-909 -> TR-909/samples (`http://machines.hyperreal.org/categories/drum-machines/TR-909/samples`). However, this same web page was visited lower than 10 times compared to over 360 visits to the former web page. This difference informs the web masters that the information delivery is not very efficient for the "categories" part.

Another observation is that although the "manufacturers" web page has both "categories" and "samples" outreach hyperlinks and in that order, people who visited manufacturers are more likely to visit samples rather than categories. This finding implies that users are more likely to browse the examples rather than the abstract "categories". Even if "categories" is put in the first place, and "samples" is in third place on the web page, the ratio of visits is 1:6 (see Figure 4.1).

4.2.1.3 Web Links Usage Distribution

When compared with the path traversal distribution as shown in Figure 4.5, the 4 web pages listed above are well visited, but the users are not browsing via the hierarchical structure. Note in Figure 4.5 that the link from "manufacturers" to "samples" is scarcely used (31 times; since it is below the threshold, it is not shown in the view). Given that this page is heavily visited (660 times) and that no links to this page within the web site contribute significantly to the number of visits, the implication is that either the web page "samples" are bookmarked by the users or they are linked by other related web sites.

Another interesting pattern is revealed when the web page visits view and hyperlink traversal view are compared. In the upper part of Figure 4.3 and Figure 4.5,

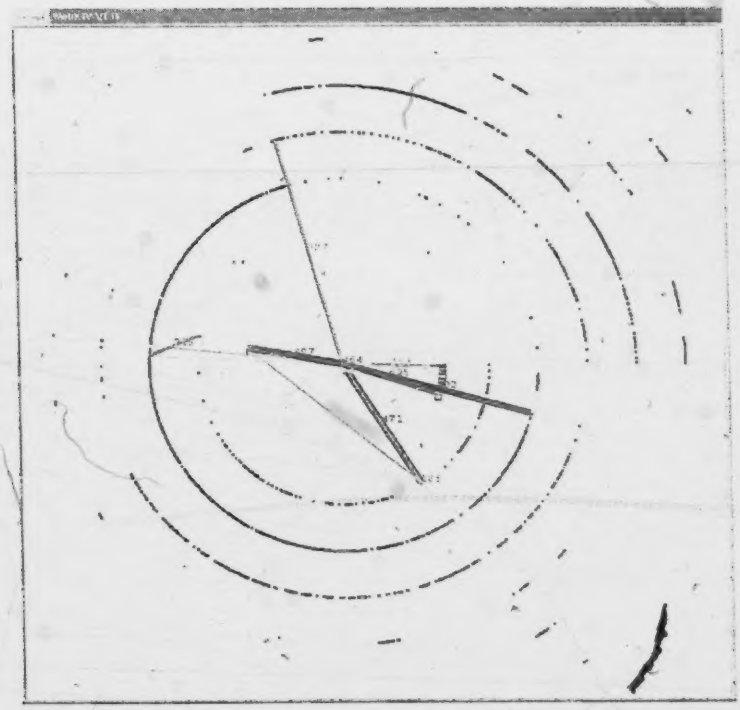


Figure 4.5: Music Machine hyperlink traversal distribution

the web page visits view shows that the page *"software/Windows"* was visited 290 times in a week (see Figure 4.3, the upper end of the line at the top of the graph), and that for 227 times the users who made these visits returned from this page to the home page (as shown in Figure 4.5, the long straight line from top to center). The finding shows that the users archived their goals without checking other information after they arrived on the web page. These figures imply that the web site structure for the information delivery was easily followed by the users.

The observations made in Section 4.2.1.2 and Section 4.2.1.3 imply that web surfers do not like the classification of the musical instruments, either because they are not familiar with the classification or the classification is unclear or unhelpful. Surfers seem to feel more comfortable with samples and brand names of the musical instruments rather than with the abstract names of classification. This conclusion suggests that a web site believed to be well-organized by web designers will not necessarily be the surfers' favorite.

Another thought based on the observation above is that hierarchy and classification are tools to help people find things more quickly, but most people are not familiar with the classification techniques and often feel confused by the names of the categories. A potential improvement is that if classification and abstraction have to be used, some well-known products and/or brand names should be added to help surfers know what the categories are referring to. For example, car sale web sites usually use "economic", "sub compact", "compact" and "full-size" categories to classify the different models of cars, but a potential customer might not be sure what kind of car is a "compact" and/or how big a "subcompact" car is. He may have to try two or more categories to find out which category fits his requirements. If the category is changed from "compact" to "Compact (e.g., Chevy Cavalier)", for example, the user will know that this category is what he wants to browse because he knows the size of the car he wants to buy is about the size of the Cavalier.

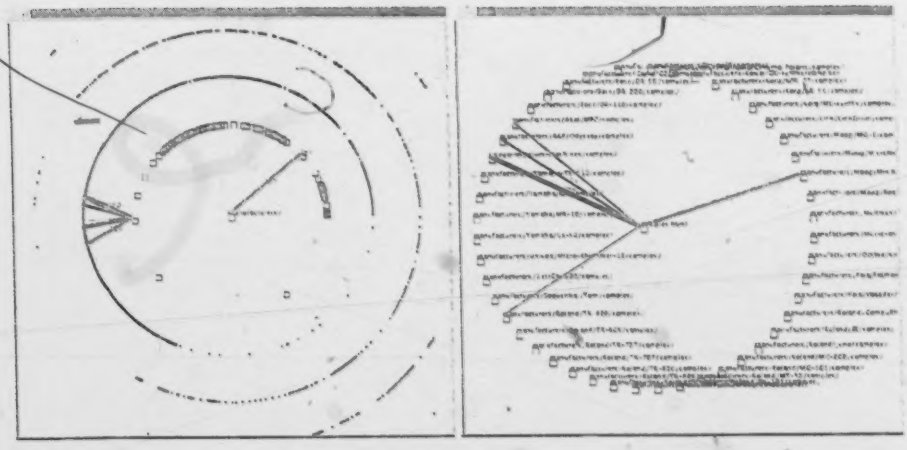


Figure 4.6: Music Machine user browsing patterns related to drum machines

The "hot information" on this web site is the information related to "drum-machines", more specifically, drum-machines made by the Roland company. In Figure 4.6, both views are generated by applying the "subtree zoom-in" operator. The root node of the left view is "manufacturers." The right red line shows the hyperlink usage from "manufacturers" to "Roland". This path is the most used among all the brand names. The "samples" on the left part of the left view was regenerated to the right view, and "samples" is the root node. A detailed study of the links usage revealed that "drum-machines/samples" and "Roland/TR-909" are the most visited web pages.

4.2.1.4 Web Usage Difference via Times

In Figure 4.7, the web usage pattern evolvement results are shown. The data are from Jan. 01, 1999 to Jan, 28, 1999. Each week, web usage logs were processed to get the usage pattern. Figure 4.7 (a) was generated by subtracting the week 1 web log data from week 2, Figure 4.7 (b) was generated by subtracting week 2 from

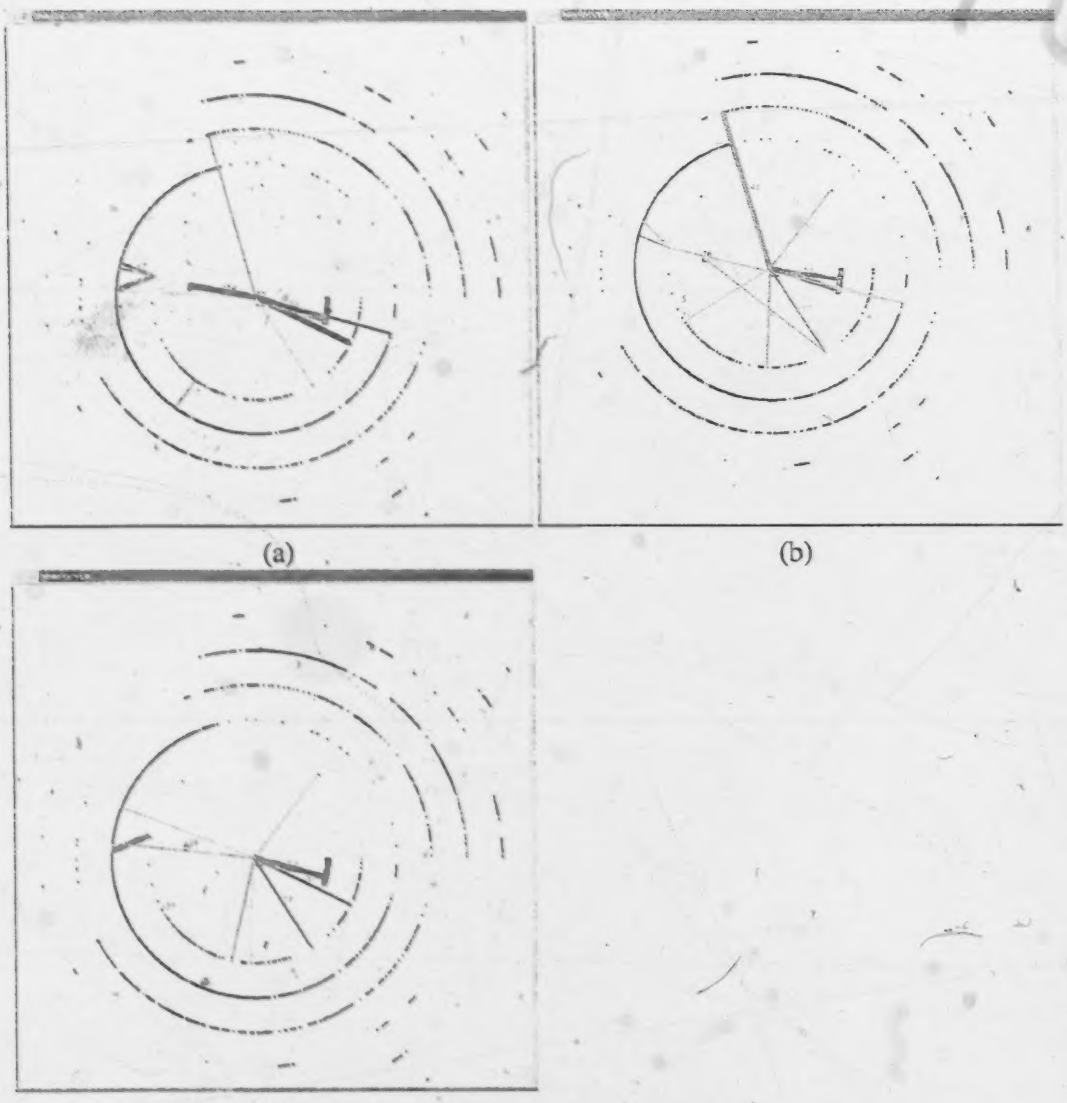


Figure 4.7: Music Machine web usage difference
Red represents positive numbers Blue represents negative results

week 3, and Figure 4.7 (c) was generated by subtracting week 3 from week 4. The threshold setting was from 20 to 50, and the red line represents a positive number after the subtraction whereas the blue line represents a negative number.

The web log data statistics are shown in Table 4.1. Except for the data for Week 1, which included the New Year's holidays, all the web log files are of roughly the same size.

In Figure 4.7 (a), most of the lines are red, which means that the subtraction resulted in a positive number. These red lines do not represent the changes in user interests, because the user visits are less in Week 1 compared with those of Week 2, likely due to the New Year's holiday.

A comparison of the second and third views reveals that the 4 sample web pages presented in Section 4.1 were visited less and less from week 2 to week 4.

4.2.1.5 NCMs Comparison

Navigational Compression Models aim at improving user navigation. By applying NCM knowledge to the web data, the surfers can find information by using a shorter traversal path. Different parameter settings of NCMs can change user traversal patterns drastically.

For example, in Figure 4.8, the two graphs show that most of the red lines representing the more traversed hyperlinks are related to the `http://machines.hyperreal.org/manufacture.html` and the homepage `http://machines.hyperreal.org/index.html`. Because these pages are visited most, the NCMs treat these web pages as the content pages which surfers want to look at, but actually they only pass through these web pages. The outer circles contain the real content of the music machines, and where the user should be suggested to go, but the graphs indicate that the outer hyperlinks (the blue lines represent the saved paths) are recommended by the NCMs as non-content pages and thus are saved.

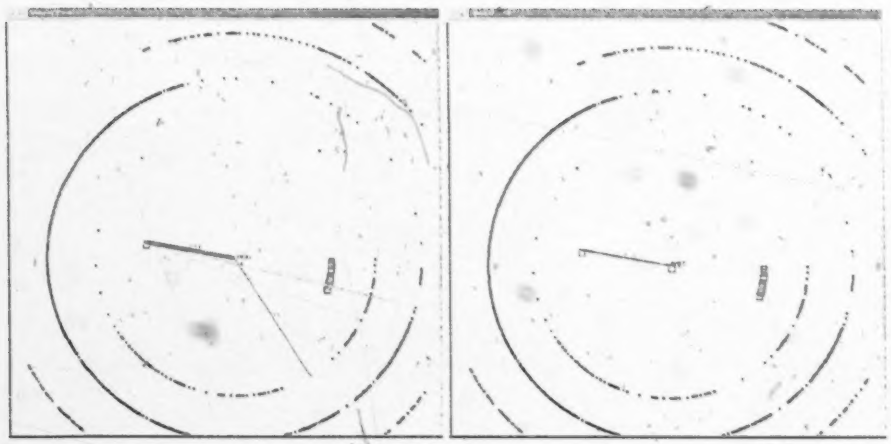
For example, at the right bottom of Figure 4.8, the web page "/Roland/Overview" and "/Roland/TR-33" are the content pages, but the NCMs think these two pages are not "important" to users.

In Figure 4.9, the content page setting criteria were changed. Not only the visit times to a web page must be considered, but also the time spent on the web page. Only the web pages that were visited more than the threshold and the times spent on each visit on the web pages that were over the threshold can be selected as content pages. Both graphs in Figure 4.9 reveal that the homepage is not the content page anymore. Although it is still the most visited web page, no more than 20 surfers (the total number of surfers is 27,115) remain at the home page for more than 1 minute. The web page `http://machines.hyperreal.org/manufacture`, however, is still treated as a content page. The reason might be that this page has more than 50 links, so users may need more time to browse the page. The obvious difference between the two graphs is that in Figure 4.9 (b), the visits to web page `http://machines.hypereal.org/categories/software` and `http://machines/hyperreal.org/categories/software/Windows` are saved. This difference indicates that the threshold setting for the right graph is higher, so the content pages of "software/Windows" are not recommended by the NCMs system.

By displaying different NCMs-generated data as well as the data generated by the same NCMs with the different parameters, the visualization system can compare the effectiveness of the web mining results and provide some insights into domain-specific adjustments.

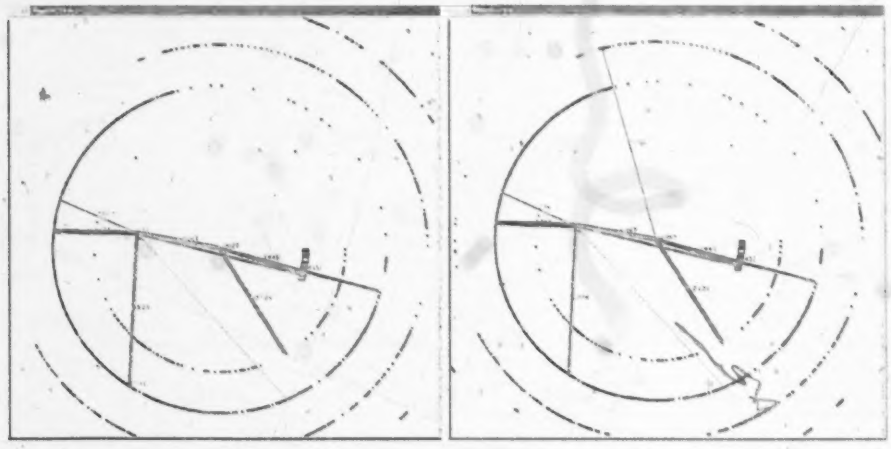
4.2.1.6 Web Surfing Animation

The data used in animation is for Jan/01/99, starting from midnight. The frame rate is set to 10 frames/second, and the animation rate is 50 times faster than real time; e.g., one second of animation equals 50 seconds of real time. Each dot represents a



(a) (b)

Figure 4.8: Music Machine NCM Comparison I
The number of surfer visits is the parameter for deciding Content pages.
(a) the 50 most visited web pages as content pages **(b)** the 100 most visited web pages as the content pages.



(a) (b)

Figure 4.9: Music Machine NCM Comparison II
The number of surfer visits and the time the surfers spend more than 1 minute is the parameter to decide content pages. **(a)** The content pages must be visited over 20 times **(b)** The content pages must be visited 200 times

user. The dot disappears if the user does not click in 30 minutes. In this experiment, all the dots are coloured as black since the user domain knowledge is unknown.[2]

When animation is started, the traversal patterns are revealed as the time lapse, as shown in Figure 4.10. Besides the changes in the usage of hyperlinks that it shows, the animation also reveals where the surfers like to stay, which pages are "go through" pages and what pages are "content" pages, on which, surfers spend more time to read the content. In Figure 4.10 (b) and Figure 4.10 (c), most of the "dots" would like to spend more time at the outer-circle pages, which contain the concrete information of the music instruments. Another interesting observation is that at the beginning of the animation, only several surfers were browsing at the same time, and the web site was quiet (Figure 4.10 (a)). Gradually, more and more surfers were browsing on the web sites. The web site community was really active, with more than two dozen surfers browsing at the same time. This finding makes sense because the time started from midnight.

4.2.2 http://www.cs.ualberta.ca/

Figure 4.12 is the web site of the Department of Computing Science of the University of Alberta. The traffic is shown in Table 4.2. Although the web site is big, compared to the Music Machine web site, it draws only 2- 3 times more visits (compare with Table 4.1). This web site provides information about programs and admissions, courses, research activities and Faculty and Graduate student information.

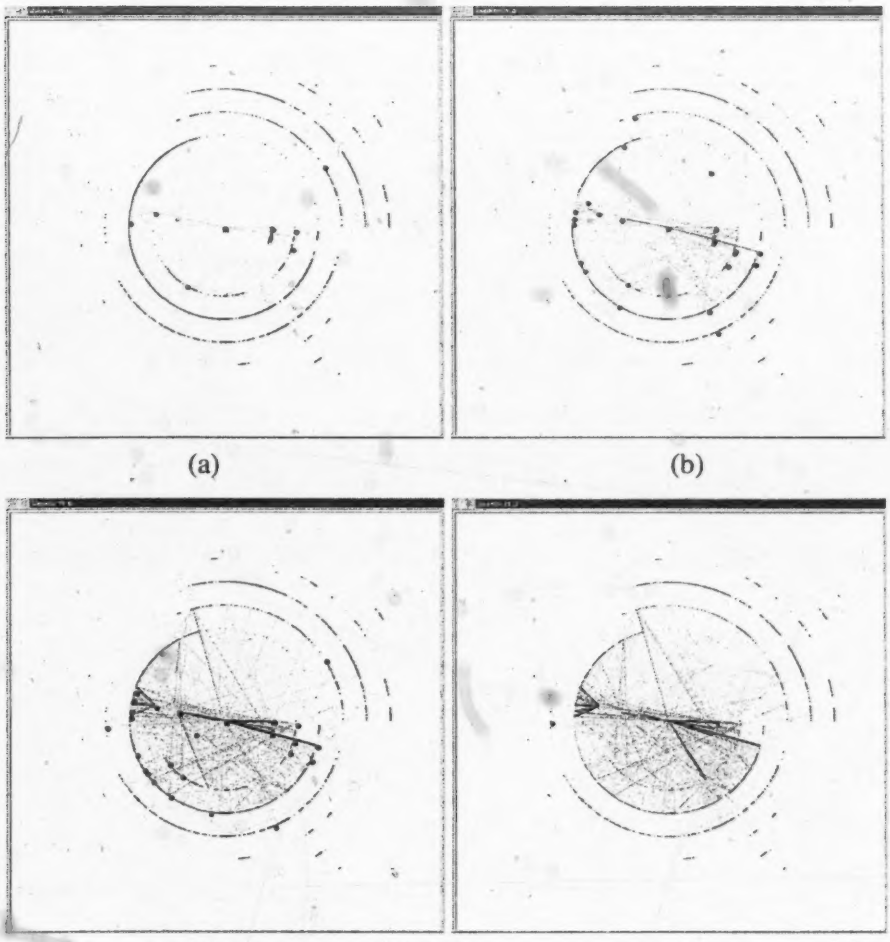


Figure 4.10: Music Machine Web Surfing Animation

Home
Search
Netscape
Print
Security
Shop
Department of
Computing Science
University of Alberta
About the Department
Programs and Admissions
Course Catalogue
Research Activities
People
Employment
Talks and Events
Alumni and Sponsors
Resources
Local Interest Only
News:
December 9: Xiang Wan is giving a Graduate Student Seminar titled Tutoring System for SPECTRUM tomorrow at 10:00 am in CSC 333.
As part of the Distinguished Lecture Series, Rah Willener, from the Technology and Manufacturing Group, Intel Corporation, is giving a lecture titled Breaking Barriers to Moore's Law tomorrow at 3:30 pm in CSC B-02.
Baochun Bai is giving a Graduate Student Seminar titled Active Feedback Multicast with Congestion Control on Wednesday December 12 at 10:00 am in CSC 333.
The Department of Computing Science is hosting KDD 2002: The International Conference on Knowledge Discovery and Data

Figure 4.11: Home page of UACS

4.2.2.1 Web Site Structure

The structure of the web site is shown in Figure 4.12. 14 levels of nodes are displayed in WebKIV. Since the web site is relatively quiet for its size, only part of the first 7 level web pages are visited (Figure 4.13). Most of the web pages located at the outer circles are personal homepages, which include many software documents, personal talk slides, etc.

4.2.2.2 User Visits Distribution

Figure 4.13 shows the "hot spots" of the web site. It shows that the following web pages are the most visited web pages:

1. `http://www.cs.ualberta.ca/courses`,

[2]Although the WebKIV System use colourful dots to represent the different domains the users come from (e.g., red for .edu, yellow for .com), the user domain information of the data sets we got is anonymous.

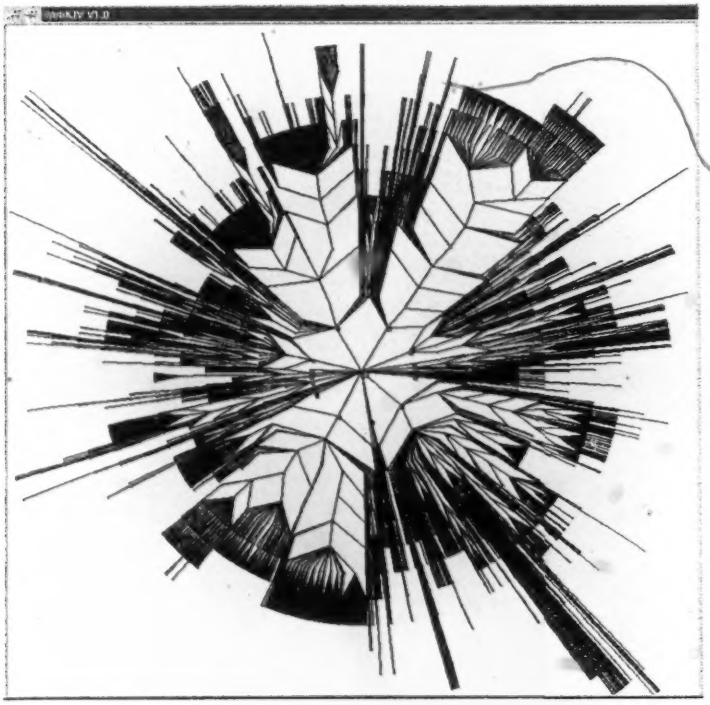


Figure 4.12: Web Site Structure of Department of Computing Science, U of Alberta(UACS) October, 2001
The depth restriction of the radial tree is set to 14

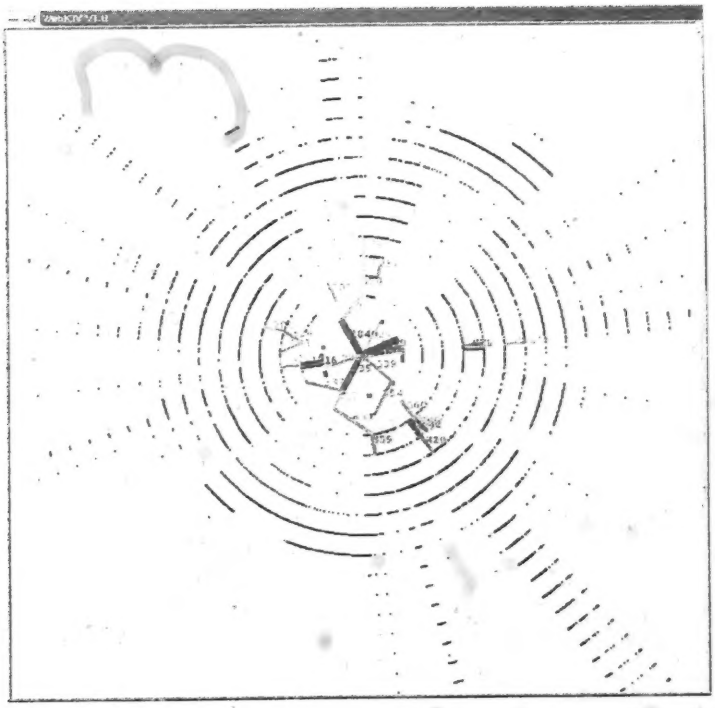


Figure 4.13: UACS Web Site visits distribution
Data: from Oct/01/01 to Oct/07/01 **Threshold:** 100-620

2. `http://www.cs.ualberta.ca/programs/`,

3. `http://www.cs.ualberta.ca/people/faculty/` and

4. `http://www.cs.ualberta.ca/~chinook`

The `http://www.cs.ualberta.ca/~chinook` was visited more than 1000 times. It is the only one web page which is not a "pass through" web page. It is unknown, however, whether its closeness to the department homepage or its content is the cause of its popularity. The rest of the most visited web pages (i.e., visited over 1000 times) are all the "pass through" pages.

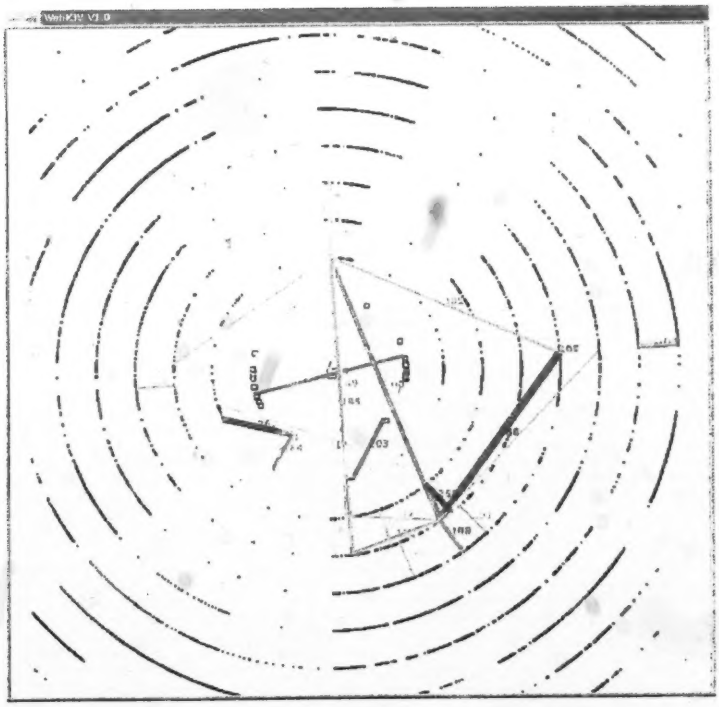


Figure 4.14: UACS Web Links Usage Distribution
Data: from Oct/01/01 to Oct/07/01 **Threshold:** 20-320

4.2.2.3 Web Links Usage Distribution

In Figure 4.14, the upper part of the graph was well visited, and from it, the surfers visited many other parts. They are some of the web pages related to courses provided by the department, and from the course description pages, most of the surfers go to the professors' web pages that relate to the courses they will teach. After general observation, the threshold was changed to reveal the most important pattern. In Figure 4.15, there is a triangle on the left right part across the radial tree. The content of the web pages are all related to a Cmput201 course, including the course description, course material from the professor, and the lab information.

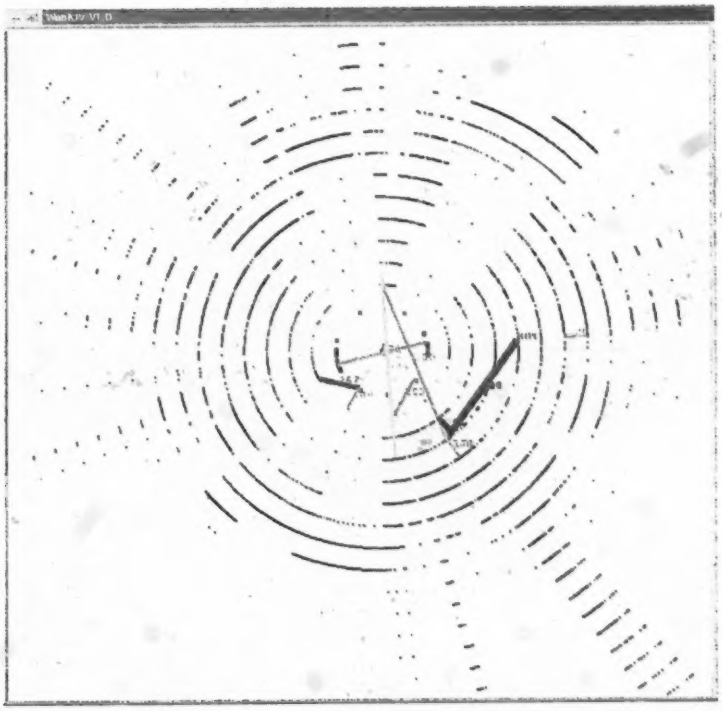
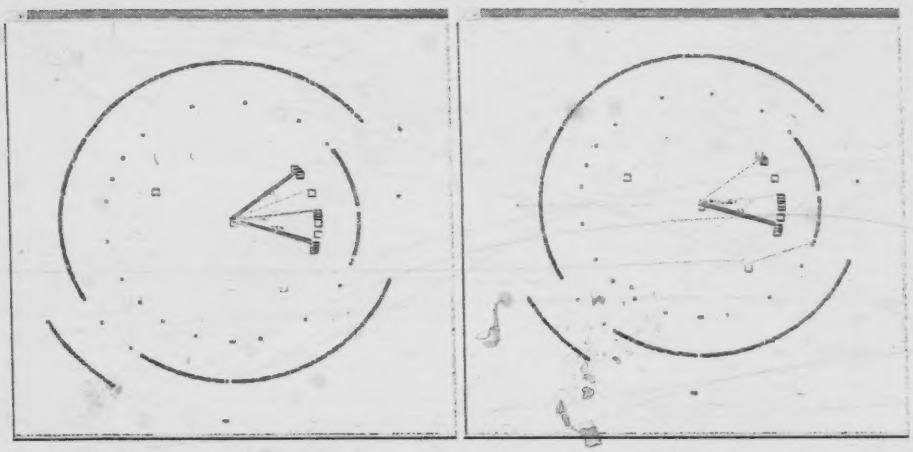


Figure 4.15: Web Links Usage Distribution
Data: from Oct/01/01 to Oct/07/01 **Threshold:** 100-320



(a) (b)

Figure 4.16: UACS Subgraph of ~chinook
(a) User Visits Distribution (b) Link Usage Distribution

4.2.2.4 Chinook - Subtree Visualization of UACS

The original graph is too big to allow for a detailed observation of the interested web data, so a subtree rooted from `http://www.cs.ualberta.ca/~chinook` was generated (Figure 4.16).

Chapter 5

Summary and Future Work

The previous chapters have discussed the WebKIV system architecture and some experiments. This chapter will discuss some technical points and trends in designing a visualization system. Some potential usages and improvements of the WebKIV system will be proposed. This chapter will end with conclusions.

5.1 Summary

5.1.1 Web Site Mapping

Although web site topology is an abstract data set without any real spatial attributes, web site visualization systems have to assign positions for each web page. The distance of the space among the web pages in visualization systems may inadvertently give the viewers the wrong perception of the relationship among the web pages. The wrong perception is generated either by the web crawling algorithm (File directory system, Breadth-first search or Depth-first search algorithm) or the hidden information (The non-tree links are not shown, for example) when the web site structure is displayed. For example, the semantically very closed web pages may be put in different positions in visualization systems. As a result, viewers might feel that these pages are irrelevant. This problem is reduced if the links among them can be displayed. Unfortunately, doing so is actually almost impossible. Be-

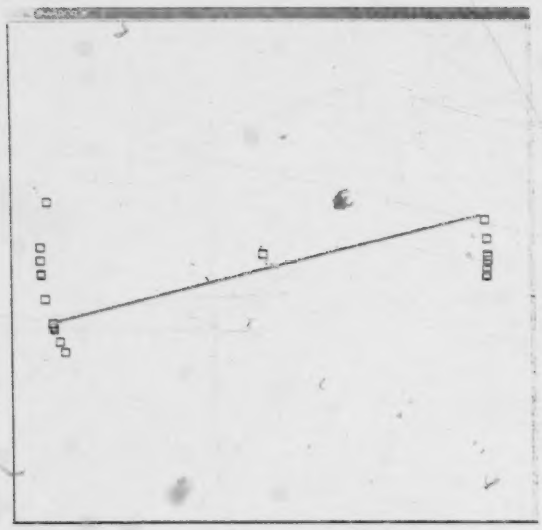


Figure 5.1: Radial Tree Problem

cause of viewability, most of the visualization systems do not show the non-tree links. A trade-off exists for the visualization system designers between viewability and performance and the completeness, especially for the large-size web sites.

Distorted magnification and 3D techniques are good at helping users understand web site navigation but not at pattern discovery. Pattern discovery always needs an overview of the web site, whereas distorted display can show only part of the structure. As the web sites get bigger and bigger, linear magnification is not enough to provide detail of the correlation of the web pages if the web pages are scattered across the radial tree (see Figure 5.1). Combined techniques must be used to provide for the future requirements. For example, for the whole web site, a Zooming and Panning technique is used. Whenever there is a need to drill down and the linear magnification cannot provide enough detail, a distorted display technique can be used for a specific area.

Large web sites can be broken down into small sections when visualized and

analyzed if the content of different sections is not tightly correlated, but the web site break down has to be very carefully made so that the loss of correlation is minimum if not zero. In the WebKIV system, "Subtree Generation" was the initial experiment. Reducing the size of the graph is very useful if the interesting patterns are within a subtree of a radial tree.

As web sites get bigger and bigger, web site mapping requires abstraction. If the web pages can be collapsed into clusters in a semantically accurate manner, the scalability is not a problem since the important patterns are always the portion of the web sites which need to be expanded.

5.1.2 Visualization System Design

For Zooming and Panning visualization systems, a multiple view is essential. The multiple view is not necessarily used to display the different attributes or the different data sets for the comparison used in [14], but is used in the WebKIV system to alleviate the challenge of context switching. It is very convenient to include or exclude some parts of the data set and to shift from one pattern to another.

5.1.3 Performance

The current version of WebKIV system can visualize 60,000 web pages. In our experiments, 58,989 nodes were visualized. Several factors were considered for performance issues in the process of the WebKIV design. First, the initial position of nodes is fixed and precomputed; the node position does not change when the window size changes. Second, usage information such as the link traversal data and well-visited web page data is also precomputed. Third, all the precomputed data are cached. This feature is memeory-consuming , but provides the necessary speed when Zooming and Panning are operated. Each time there is a Zooming and Panning operation, the selected subset of nodes and lines information is recorded in

a buffer. When a further operation is made, the system always checks if the newly selected nodes and lines are included in the buffer. Since the subset of the data in the buffer is smaller, they process faster. Only when the chosen nodes and lines are not all included in the buffer, will the system retrieve the nodes and lines information from the initial tables. This feature is efficient when a large graph is involved. For example, to display the 60,000 web page visits statistics requires more than 1 second to process and display all the information, but the time to display thousands of nodes visits information chosen from less than 10,000 nodes is negligible[1]. For pattern discovery, viewers commonly first select a relatively large area which includes all the patterns, then choose some "important" patterns that they are interested in and drill down. After the viewers finish investigating the specific patterns, no matter whether information they have found is useful or not, they go back to the original states to either choose other parts of the graph to start drilling down again or to select the display of other data attributes to explore.

5.2 Future Work

Future work with the WebKIV system can occur in two areas: visualization techniques which improve the system usability and potential new usages.

5.2.1 Visualization Algebra

The high level conception of WebKIV as web data types and visualization operators was never really exploited, except to guide the design of the prototype and in the explanation of some of the multi-layer visualizations.

Some of the original discussions on which this work was based could be steered in the direction of developing a more formal visualization algebra, where the data

[1]The time has only a relative meaning since the different hardware and the different date sets can vary the time.

types are more precisely defined, and the combination of operators is investigated to see whether they can be constrained by interesting algebraic properties. For example, we have already seen that the link-based representation of page counts can be combined with a similar representation of hyperlink traversals, so that the subtraction operator helps identify web entry points (e.g., with book marks, direct typing by URLs, etc.). A more general visualization algebra could capture a much broader set of visualization properties, and provide the basis for an interactive web visualization interpreter.

5.2.2 Fisheye Distortion

As was discussed in Section 5.1.1, the disktree has difficulty showing the detailed information for a pattern across the radial tree. Although the radial tree algorithm ensures that all the web pages have the shortest distance from the root page, two web pages hyperlinked directly might be put very far apart in the radial tree. The fisheye distortion technique is a good way to solve this problem. A fisheye effect can be achieved by using a simple geometric transformation instead of regenerating the whole graph [27].

5.2.3 Content Information vs. URLs

The present WebKIV system uses URL as the web page title, which reveals little content information for the web pages. Another way is to use the meta data of the web pages to represent the web pages, for example, the key words extracted or mined from the web pages, or the important sentences (e.g., the title of the web pages).

5.2.4 Surfer Navigation Assistant

Although the WebKIV system is designed for use offline and aimed to help web masters to manage the web site and web researchers to analyze the web patterns, it can also be used as a web surfer navigation assistant. It can show the web site statistics such as which parts of the web site are "hot spots" and where the new added information is. As discussed in Section 2.3.2, a distorted display is more suitable for helping individual surfing because only a fraction of the information needs to be computed and displayed. Our system works well for small web sites as a web navigation assistant, but for the web sites which contain more than several hundred web pages, the abstraction technique has to be used to keep the computation time negligible.

5.2.5 Other Potential Applications

Compared with other visualization systems, which usually visualize mining knowledge, web site structure and web content information separately, the WebKIV system integrates all three data sets together. The WebKIV system has many potential usages. For example, the WebKIV system can visualize the web page clustering data generated either by content based clustering algorithms or by web usage clustering methods. Beyond merely comparing the difference of the two mining results, the system can use one data pattern to evaluate another. For example, web surfer clustering can be shown side by side with web page clustering to evaluate or validate the accuracy of the web page clustering algorithm.

5.3 Conclusion

In this dissertation, a web data visualization system (WebKIV) has been presented, several design principles have been discussed and implemented, and experiments have been made and explained. The system aims at showing how a web site is

organized as well as how the web site was used by the users. The goal is to help web researchers to understand the correlation between web usage and web site design. Besides visualizing the statistic patterns of the web site usage, web site structure and the content of the web pages, WebKIV provides dynamic animation of the web user behaviour with aggregate navigation path changes over different time periods.

The *multiviews* feature alleviates the problems of context switching and the context loss when drilling down. Animation functionality provides more realistic web user activities so that the pass through or the content pages can be easily identified. Background noise is eliminated to a minimum by threshold settings. "Subtree Zooming" is novel to the web visualization community, and provides not only a kind of zoom function but also the possibility of breaking down the large web site into smaller sites. It also provides a platform to make multiple view comparison possible. For the web usage visualization, the WebKIV system is effective for validating and correlating the different data sets. The present study is the first to use web-mining generated web data to compare with the original data so that the difference and the effect can be observed and manipulated.

Bibliography

[1] http://www.iguide.co.il/maps/6.html.

[2] http://www.december.com/web/text/images/cyberland.gif.

[3] http://www.inventix.com.

[4] http://www.mercuryinteractive.com.

[5] Tower maps. http://www.towermaps.com/product_summary.htm.

[6] Keith Andrews. Visualizing cyberspace: Information visualization in the harmony internet browser. In *Proceedings of First IEEE Symposium on Information Visualization*, pages 97–104, October 1995.

[7] Eric Z. Ayers and John T. Stasko. Using graphic history in browsing the world wide web. In *Proceedings of the Fourth International World Wide Web Conference*, pages 11–14, Boston, December 1995.

[8] Michelle Q. Wang Baldonado and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of Advanced Visual Interfaces*, pages 110–119, May 2000.

[9] Lyn bartram, Albert Ho, John Dill, and Frank Henigman. The continuous zoom: A constrained fisheye technique for viewing and navigation large information spaces. In *Proceedings of UIST'95*, pages 207–215, November 1995.

[10] Bettina Berendt. Understanding web usage at different levels of abstraction: coarsening and visualising sequences. In *Proceedings of WEBKDD 2001 - Mining Log Data Across All Customer TouchPoints*, San Francisco, CA, Auguest 2001.

[11] Igor Cadez, David Heckerman, and Christopher Meek. Visualization of navigation patterns on a web site using model_based clustering. In *Proceedings of KDD 2000*, pages 280–284, 2000.

[12] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of CHI'91*, pages 181–188, 1991.

[13] Ed H. Chi, Peter Pirolli, and James Pitkow. The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a web site. In *Proceedings of CHI 2000 Human Factors in Computing Systems*, 2000.

[14] Ed.H. Chi and Stuart K. Card. Sensemaking of evolving web sites using visualization spreadsheets. In *Proceedings of the Symposium on Information Visualization*, pages 18–25, San Francisco, California, 1999.

[15] E.H. Chi, J. Pitkow, J. Mackinlay, P. pirolli, and J. Konstan. Visualizing the evolution of web ecologies. In *Proceedings of CHI'98*, pages 400–407, 644–645, 1998.

[16] Andy Cockburn, Saul Greenberg, Bruce McKenzie, Michael Jasonsmith, and Shaun Kaasten. Webview: A graphical aid for revisiting web pages. In *Proceedings of the OZCHI'99 Australian Conference on Human Computer Interaction*, November 1999.

[17] Andy Cockburn and Steve Jones. Design issues for world wide web navigation visualization tools. In *5th RIAO Conference: Computer-Assisted Information Searching on the Internet*, pages 55–74, Montreal, Canada, June 25-27 1997.

[18] R. Cooley, B. Mobasher, and J. Srivastava. Grouping web page references into transactions for mining world wide web browsing patterns. In *Proceedings of IEEE Knowledge and Data Engineering Workshop*, pages 2–9, 1997.

[19] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.

[20] Robert Walker Cooley. *Web Usage Mining Discovery and Application of Interesting Patterns from Web Data*. PhD thesis, University of Minnesota, May 2000.

[21] Tom Downes and Shane Greenstein. Universal access and local commercial internet markets. http://www.kellogg.nwu.edu/ faculty/greenstein/ images/htm/Research/isppap4.pdf, June 1998.

[22] Usama Fayyad, Georges G. Grinstein, and Andreas Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2001.

[23] G.W. Furnas. Generalized fisheye views. In *Proceedings of CHI*, pages 16–23, 1986.

[24] N. Gershon, L. Rosenblum, R.A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, and D. Thalmann. From perception to visualization. In *Scientific Visualization - Advances and Challenges*, pages 129–139. Academic Press, London, 1994.

[25] Nahum Gershon, Stephen G. Eick, and Stuart Card. Information visualization. In *ACM SIGGRAPH Course notes 8*, August 1996.

[26] R.J. Hendley, N. S. Drew, A. M. Wood, and R. Beale. Narcissus: Visualizing information. In *Proceedings of IEEE Information Visualization Symposium*, pages 90–97, Los Alamitos, 1995.

[27] I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. In *IEEE Transactions on Visualization and Computer Graphics*, volume 6 of *1*, pages 24–43, 2000.

[28] M. Hersovici, M. Jacovi, Y.S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur. The shark-search algorithm - an application: Tailored web site mapping. In *Proceedings of the 7th International World Wide Web Conference*, pages 317–326, 1998.

[29] Ron R. Hightower, Laura T. Ring, Jonathan I. Helfman, Benjamin B. Bederson, and James D. Hollan. Graphical multiscale web histories: A study of padprints. In *Proceedings of ACM Hypertext 98 Conference*, pages 58–65, 1998.

[30] Heike Hofmann, Arno Siebes, and Adalbert Wilhelm. Visualizing association rules with interactive mosaic plots. In *the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 227–235, August 2000.

[31] Kasper Hornbak and Erik Frokjar. Reading of electronic documents: The usability of linear, fisheye, and overview+detail interfaces. In *Proceedings of ACMCHI*, pages 293–300, March 2001.

[32] Thorsten Joachims, Dayne Freitag, and Tom Mitchell. Webwatcher: A tour guide for the world wide web. In *Proceedings of IJCAI'97*, pages 770–777, August 1997.

[33] Nielson Jokob. The art of navigating through hypertext. *Communications of the ACM*, 33(3):296–310, 1990.

[34] Paul Kahn and Krzysztof Lenk. *mapping web sites*. RotoVision SA, 2001.

[35] T. Alan Keahey and Edward L. Robertson. Techniques for non-linear magnification transformations. Technical report, Indiana University, 1996.

[36] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408, New York, May 1995.

[37] J.H. Larkin and H.A. Simon. Why a diagram is (sometimes) worth than thousand words. *Cognitive Science*, 11(1):65–99, 1987.

[38] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. Denim: Finding a tighter fit between tools and practice for web site design. In *Proceedings of CHI'2000*, pages 510–517, April 2000.

[39] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *International Conference on Knowledge Discovery and Data Ming*, pages 125–134, San Diego, CA, August 1999.

[40] Sarkar M and M.H. Brown. Graphical fisheye views of graphs. In *Proceedings of ACM SIGCHI*, pages 83–91, 1992.

[41] Yoelle S. Maarek, Michal Jacove, Menachem Shtalhaim, Sigalit Ur, and Dror Zernik. Webcutter: A system for dynamic and tailorable site mapping. In

Proceedings of WWW6 Conference, pages 713–722, Santa Clara, California, April 1997.

[42] Nelson Minar and Judith Donath. Visualizing the crowds at a web site. In *Proceedings of CHI'99*, 1999.

[43] Bamshad Mobasher, R. Cooley, B. Mobasher, and J. Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop KDEX'99*, pages 19–25, November 1999.

[44] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. Technical Report TR99-010, Department of Computer Science, Depaul University, 1999.

[45] Sougata Mukherjea and James D. Foley. Visualizing the world-wide web with the navigational view builder. *Computer Networks and ISDN Systems*, 27(6):1075–1087, 1995.

[46] Sougata Mukherjea and Yoshinori Hara. Focus+context views of world-wide web nodes. In *Proceedings of the eighth ACM conference on Hypertext*, pages 187–196, 1997.

[47] Tamara Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. In *IEEE Symposium on Information Visualization*, pages 2–10, October 1997.

[48] Tamara Munzner. Drawing large graphs with h3viewer and site manager. In *Symposium on Graph Drwaing*, pages 384–393, August 1998.

[49] Tamara Munzner. *Interactive Visualization of Large Graphs and Networks*. PhD thesis, Stanford University, June 2000.

[50] Tamara Munzner and Paul Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. In *Proceedings of VRML'95*, pages 33–38, San Diego, California, December 1995.

[51] Hayato Ohwada and Fumio Mizoguchi. Integrating information visualization and retrieval for discovering intenet source. In *Discovery Science, Third International Conference*, pages 38–49, December 2000.

[52] James E. Pitkow and Krishna A. Bharat. Webviz: a tool for world-wide web access log analysis. In *Proceedings of First International WWW Conference*, pages 271–277, May 1994.

[53] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Communications of the ACM, 34(2): 189-194*, February 1991.

[54] R. Stout. *Web Site Stats: Tracking Hits and Analyzing Traffic*. McGraw-Hill, 1997.

[55] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1994.

[56] Edward R. Tufte. *Visual Explanations*. Graphics Press, 1997.

[57] Alan Wexelblat. History-based tools for navigation. In *Proceedings of the Hawaii International Conference On System Science*. IEEE Computer Society Press, January 1999.

[58] Alan Wexelblat and Pattie Maes. Footprints: History-rich tools for information foraging. In *Proceedings of CHI'99*, pages 270–277, May 1999.

[59] Graham J Wills. Nicheworks - interactive visualization of very large graphs. *Journal of Computational and Graphical Statistics*, 8(2):190–212, 1999.

[60] Tong Zheng, Yonghe Niu, and Randy Goebel. Webframe: In pursuit of computationally and cognitively efficient web minging. In *Proceedings of the 6th Pacific-Asia Conference, PAKDD*, pages 264–275, May 2002.